Georgia State University
ScholarWorks @ Georgia State University

Computer Science Dissertations | Department of Computer Science

Summer 7-12-2017

New data analytics and visualization methods in personal data mining, cancer data analysis and sports data visualization

Lei Zhang

Follow this and additional works at: https://scholarworks.gsu.edu/cs_diss

Recommended Citation
Zhang, Lei, "New data analytics and visualization methods in personal data mining, cancer data analysis and sports data visualization." Dissertation, Georgia State University, 2017.
https://scholarworks.gsu.edu/cs_diss/126

This Dissertation is brought to you for free and open access by the Department of Computer Science at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Computer Science Dissertations by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact scholarworks@gsu.edu.

New data analytics and visualization methods in personal data mining, cancer data analysis and sports data visualization

by

Lei Zhang

Under the Direction of Ying Zhu PhD

ABSTRACT

In this dissertation, we discuss a reading profiling system, a biological data visualization system and a sports visualization system. Self-tracking is getting increasingly popular in the field of personal informatics. Reading profiling can be used as a personal data collection method. We present UUAT, an unintrusive user attention tracking system. In UUAT, we used user interaction data to develop technologies that help to pinpoint a users reading

region (RR). Based on computed RR and user interaction data, UUAT can identify a readers reading struggle or interest. A biomarker is a measurable substance that may be used as an indicator of a particular disease. We developed CancerVis for visual and interactive analysis of cancer data and demonstrate how to apply this platform in cancer biomarker research. CancerVis provides interactive multiple views from different perspectives of a dataset. The views are synchronized so that users can easily link them to a same data entry. Furthermore, CancerVis supports data mining practice in cancer biomarker, such as visualization of optimal cutpoints and cutthrough exploration. Tennis match summarization helps after-live sports consumers assimilate an interested match. We developed TennisVis, a comprehensive match summarization and visualization platform. TennisVis offers chart-graph for a client to quickly get match facts. Meanwhile, TennisVis offers various queries of tennis points to satisfy diversified client preferences (such as volley shot, many-shot rally) of tennis fans. Furthermore, TennisVis offers video clips for every single tennis point and a recommendation rating is computed for each tennis play. A case study shows that TennisVis identifies more than 75% tennis points in full time match.

INDEX WORDS: Reading Profiling, Self-Tracking, Cancer Biomarker, Sports Highlight Summarization, Audio Signal Processing

New data analytics and visualization methods in personal data mining, cancer data analysis and sports data visualization

by

Lei Zhang

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2017

Copyright by
Lei Zhang
2017

New data analytics and visualization methods in personal data mining, cancer data analysis and sports data visualization

by

Lei Zhang

Committee Chair: Ying Zhu

Committee: Saeid Belkasim

Yanqing Zhang

Yi Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2017

DEDICATION

This dissertation is dedicated to my family : my parents, my wife and my kids.

To my dear parents, who have showered me with love for my whole life, thank you for supporting me all the way through!

To my lovely wife, Ly Le, who lashed me from my back when I was slacking. Earning your support, trust and love is my lifelong pride!

To my kid(s), Alex, Ellie and Frederic, you are so cute!

ACKNOWLEDGEMENTS

This dissertation is written under directions from Dr.Ying Zhu. I would like to take this opportunity to thank Dr.Zhu for his help throughout my Ph.D study.

I would also express my gratitude to the members of my dissertation committee, Dr. Yi Zhao, Dr. Saeid Belkasim, and Dr. Yanqing Zhang, for their advice and their valuable time spent in reviewing the material.

To myself, thanks to Zhang's curiosity of the world and his good luck, he is able to finish this dissertation (and many other achievements). But by no means this is the end of your endeavor, there is still a long way to go to change the world. "Stay Hungry. Stay Foolish."

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION AND MOTIVATION

In this dissertation, we present three research work.

UUAT is an interactive data collection technique for a person′s daily reading activity. UUAT is also a data analysis application, which helps to compute a user′s behavioral reading region. The data UUAT collected can be used for further analysis, such as data mining and knowledge discovery.

CancerVis is a comprehensive data visualization and exploration platform for biological data analysis. Besides common visual data exploration functions, CancerVis offers a data mining module, namely CutPointVis, which helps to visually determine optimal cutpoints in biomarker research.

As a sports visualzation platform, TennisVis presents a tennis match with both chart graphs and video clips. A client can obtain match statistics with charts and do query according to his own perferences. Moreover, video clips of every single tennis point are offered for a client to explore the details of a match.

1.1 Reading Profiling

Personal informatics [1, 2] has been proposed as a complement to commercial and business data mining. Commercial data mining [3] often aims to discover new knowledge to improve business opportunity or marketing strategies, while personal informatics is proposed for self analysis and personal improvement. In personal informatics, a user′s behavioral data is collected by a neutral utility. The collected data is under the user′s full control. The user can choose third-party tools or cloud based services to analyze his personal data, aiming to improve individual productivity or life quality. In addition, each person′s behavioral data may be compared with data from other users to find correlations among the data of users

with similar interests.

The first step of personal informatics is to collect detailed personal data. In recent years, self-tracking [4–6] is getting increasingly popular in personal informatics as data collection methods. Self-tracking has been used for collecting personal physical data, such as weight, heartbeat, diet, and exercises. The pervasive data collection method is manual logging. For example, a self-tracker records his weight before going to bed everyday. However, personal informatics can be applied to a much broader range of data than physical data. The data collection method should be automatic and unintrusive to meet the requirement of good scalability.

Collecting a user′s daily reading information can help understand his personal reading patterns, such as the evolution of the user′s interest over time, subject distribution, reading material complexity, and personal vocabulary list. Such information can help a user diversify his reading subjects, increase reading material complexity, or expand vocabulary. Other information, such as the time spent on different parts of an article or repeated reading of a paragraph, may also be used to discover patterns whose application may be unknown but could be useful in the future. As pointed out in [7] "For many self-trackers, the goal is unknown".

To collect such personal data, an automatic and unintrusive method is needed so that it does not interfere with the users daily reading activities. We developed a solution to identify which part of a text a user actually read and other reading related behaviors, such as reading speed and page scrolling patterns. Our developed technique, namely Unintrusive User Attention Tracking (UUAT), collects a user′s interaction data during his reading, computes user′s behavioral reading region, and provides details of that reading session. The data collected with this technique enables future personal informatics applications. To the best of our knowledge, this is a novel method that accurately identifies the user′s preferred reading region in an unintrusive way.

1.2 Data Visualization in Cancer Biomarker Research

In current cancer biomarker research, data visualization is mainly utilized to present research results. It is underutilized, though, as a tool for early biomarker research, where it can assist in data exploration and pattern discovery. Previous work has shown that interactive visualization techniques can be useful tools for early stage of dataset exploration. For example, PRISMA [8] and exploRase [9] can help researchers get an fast, intuitive, and comprehensive understanding of a new dataset. VizRank [10] helps find simple, interpretable data projections that include only a small subset of genes yet do clearly differentiate among different cancer types. These solutions are either lack of interactivity or focus only on the exploration stage.

We developed CancerVis, an interactive explanatory platform for cancer biomarker research. CancerVis helps a researcher explore high-dimensional datasets with techniques such as scatter plot, parallel coordinates and color coding. CancerVis is extensible and scalable because it is designed as a web service. In CancerVis, a dataset can be visualized from different perspectives, according to the user customization. A new chart can be added to the clients viewport and any existing chart can be removed from current viewport. Thus, this allows only useful plots to be kept in the user′s viewport. CancerVis automatically detects data ranges and data types for its given dataset. Visualization is synchronized among all plots in the viewport: selection of entries in one plot results in highlights of corresponding entries in the remaining plots. Furthermore, CancerVis tackles survival analysis by providing CutPointVis, which enables a fast, convenient and user adjustable cutpoint determination.

1.3 Visualizing a Tennis Match with On-demand Video Replay

Currently, sports analysis can be mainly divided into following categories: object detection and tracking [11–13], semantic analysis (scene analysis and event detection) [14–16] and highlight summarization [17, 18]. Object detection and tracking are used mainly for sports performance measurement. Highlight summarization is popular in sports entertain-

ment since it offers a fast and compact version of full match video for those sports fans who do not have enough time to watch live casts.

Semantics analysis in sports research refers to the extraction of video cues which can both reflect the structure of the video and be consistent with human understanding. Domain-specific knowledge are used in the semantic analysis.

Extracted semantic events maybe trivial, since the semantic events are mainly defined in domain-specific knowdege, such as sports rules. For example, in a soccer game, these semantic events include fouls, corners, penalties, goal, subsitution etc. As a sports fan, watching all these trival video clips may be as time-consuming as watching the whole replay video. [17] proposed highlight catagorization that evaluates importance of a highligth and rank all the extracted highlights. This solution detects the slow-speed replayed part in the live video. [19] uses arousal level in replayed parts to rank highlights. Arousal level is calculated by audio energy and motion activity in corresponding events. [20] detects event in sports video through audio cues. It correlates audio sigals with video frames, so that more accurate devents can be detected.

In order to present a tennis match, a platform which can both present match statistics and offer match detail (video clips of tennis points) according to user interaction, can solve the aforementioned problem.

We developed TennisVis, a chart-based highlight extraction platform for tennis matches. With TennisVis, a user can get an overview (statistics) of a tennis match at first sight. At the same time, match details (video clips of each tennis point) are offered for user selective video play. Furthermore, TennisVis offers highlight recommendation, which presents a compact compile for top rated plays in a match.

The data input of TennisVis is twofold, a Shot-by-Shot textual description (S2STD) and a full match video. Match facts can be extracted and produced through text mining the S2STD file. In order to identify a specific tennis point from a full time match video, the timing of each tennis point needs to be computed. Since timing information (time of each tennis point in the match video) is not available in S2STD, an Audio-based Tennis Rating

Framework (ATRF) is developed to extract timing information of each single tennis point and evaluates a rating of each tennis play. The extracted video clips and ratings can be used for user on-demand video highlight play or automatic highlight recommendation.

Compared to other state-of-art work, our work distinguishes itself with following points:

1. We provide a straightforward platform for a user to visualize a tennis match, match facts are presented within a single graphical user interface.

2. Although one of our goals is extraction of video highlights, our solution does not involve any vision / image processing technique. This makes our solution computationally affordable.

3. Our solution offers an on-demand video play of match highlights, a user is able to choose to view any video highlight according to his individual interest.

4. TennisVis offers automatic highlight recommendation based on text mining and audio analysis.

Chapter 2

DAILY READING PROFILING

2.1 Introduction

Personal informatics [1, 2] has been proposed as a complement to commercial and business data mining. While commercial data mining [3] often aims to discover new knowledge to improve business opportunity or marketing strategies, personal informatics is proposed for self analysis and personal improvement. In personal informatics, a user′s behavioral data is collected by a neutral utility, and it is under the user′s full control. The user can choose third-party tools or cloud based services to analyze his personal data, aiming to improve individual productivity or life quality. In addition, each person′s behavioral data may be compared with data from other users to find correlations among the data of users with similar interests.

The first step of personal informatics is to collect detailed personal data. In recent years, self-tracking [4–6] is getting increasingly popular in personal informatics [2]. For example, self-tracking has been used for collecting personal physical data, such as weight, heartbeat, diet, and exercises. The main logging method is manual logging. A self-tracker records his weight before going to bed everyday. However, personal informatics can be applied to a much broader range of data than physical data and users prefer automatic and unintrusive self-tracking methods.

In this research, we view a user′s daily online reading as a source of personal data. It should be noted that we consider a single reading activity to be relatively long and an article may span multiple screen pages.

Collecting a user′s daily reading information can help understand personal reading patterns, such as the evolution of the user′s interest over time, subject distribution, reading material complexity, and personal vocabulary list. Such information can help a user diver-

sify his reading subjects, increase reading material complexity, or expand vocabulary. Other information, such as the time spent on different parts of the text or repeated reading in a paragraph, may also be used to discover patterns whose application may be unknown but could be useful in the future. For example, As pointed out in [7] "For many self-trackers, the goal is unknown.".

To collect such personal data, an automatic and unintrusive method is needed so that it does not interfere with the user′s daily reading activities. In this research, we have developed such a solution. Our goal is to identify which part of a text a user has read and other reading related behaviors, such as timing and page scroll patterns. We developed a browser-based attention tracking technique, namely Unintrusive User Attention Tracking (UUAT). UUAT collects a user′s interaction data during his reading, computes a user′s behavioral reading region, provides details of that reading activity. The data collected with this technique enables future personal informatics applications. To the best of our knowledge, this is a novel method that accurately identifies the user′s preferred reading region in an unintrusive way.

2.2 Related Work

Self-tracking is well researched in its literature [5,6,21–23]. The practice of self-tracking refers to collection of personal data periodically. A later analysis of collected data may follow to produce statistics and other data relating to habits, behavior even feelings. The current state-of-art personal data mainly includes personal physical data, such as weight, glucose, heartbeat, sports, food/medication consumption. However, the existing pervasive methods of personal data collection are manual collection, such as the methods adopted by PatientsLikeMe [24], in which a user/patient input his biological data into his computer (usually a spreadsheet) and save it for future use. Mobile and wearable devices [5, 6, 22, 23] are also available to assist the raw data collection task. Our research adopts an user-unaware, automatic method to collect a users personal data.

Our first research goal is the computation of a users preferred reading region, the lit-

erature in this research field can be divided into two main categories. Eye-tracking based research and mouse/keyboard tracking based research. [25] is a typical eye-tracking based research. The author used eye-tracking to conclude that a users attention region and cursor are closely related, both of them can be modeled as a normal distribution. We argue that eye-tracking based methods are also affected by the initial calibration, inconvenience and its cost. On the one hand, eye-tracking device may affect a user′s behavior, furthermore, the initial calibration of device and user movement limitation make the eye-tracking method unstable. On the other hand, it is impractical to popularize an eye-tracking based application to a large scale, especially when a pervasive self-tracking is referred.

Compared to the eye-tracking research [26–28], our research is closer to the work of [29–33], which adopted interactive user-unaware methods to track a user′s browsing/reading activity. The work in [32] developed a vertical heatmap bar for long webpage viewers. It computes the dwell time of each specific part of web page. This work collects similar data as ours (user scroll), but it aims at a different goal: to navigate user during his reading. The work in [34] developed well-designed web page structure to collect user mouse interaction with specific webpage parts. Then it evaluated user attention area on the designed page. Our work is different from [32, 34] in that we do not intend to assist a users reading process (navigation or improve reading efficiency). We aim to develop an "observer", which can collect objective and accurate user reading behavioral data.

The authors of [25] collected user mouse data and synchronized eye gaze data. They proved that the user mouse data can indicate user behavior, such as which part of the screen is more attractive. The user distraction behavior can be detected and the user experience can be predicted by mouse analysis in an accuracy of 80%. The result of this research highly depends on the eye-gaze data. The research of [34] was conducted for a software usability test. They collected user data in an unaware way, which is more objective. The collected data is very detailed, which might be used to construct a "playback" for a testers tryout. The downside of this research is that it needs to log every interaction between a user and a web component (not a web page, but each HTML components in that webpage), such as

how long it took the subject to first click on a specific radio button on a page. This makes it a heavy traffic data collection application. Another similar research [35] was conducted to study how to log user interaction with Ajax-based web components. Since the research goal is to evaluate software usability, the work in [34, 35] collected raw interaction data and in a specific way. Furthermore, extra server-side support is needed to conduct the data collection, making it less scalable. The research of [30] was conducted for evaluation of search engine result pages. It collected data to discover which search engine result item is more attractive to a user. In [30], the author proposed a "viewport" technology, which blurs all the other result items except one item to force the user can put his attention on the unblurred result item. In [31], the author utilized mouse trajectory instead of the mouse click. The reason is that the mouse is more often to move on the screen than a mouse is clicked, although the mouse move does not always trigger an event that can be logged. With the comparison of eye-gaze data, they disclosed that the mouse trajectory can be used to infer a users interest on the search engine result items.

We distinguish our research from existing works as follows: our data collection is tagged as scalable and unintrusive. Furthermore, we developed a new computer human interaction technology to accurately infer a reader′s preferred reading region on the screen. This technology may pave the way for future applications in personal informatics.

2.3 Where Do You Read?

To understand online reading in an user-unaware way, it is necessary to determine a user′s Reading Region(RR). A Reading Region is defined as the screen area where a user′s reading attention effectively covers. Only those texts in the RR are possibly read and learned by a user. In this section, we will first model the RR and introduce our method to compute behavioral RR(BRR) and realtime RR(RRR).

2.3.1 Model of Reading Region

Psychologically, felt involvement [36] influences attention and comprehension. Felt involvement is low when a person starts reading any text. It increases rapidly as reading continues until felt involvement reaches a peak value. After that, felt involvement decreases due to a decrease in the attentional resources. The hypothesis in [36] can also be supported by [25].

The previous research on user attention tracking [25] has disclosed some facts when a user is doing daily reading on long articles and documents. Vertically, most of the time, the reading progresses towards the end of the screen in a line-by-line manner. However, the reading does not start from the first line on the screen to the last line of the screen. Generally, a user has a preferred reading region on the screen, which in this paper, we name it as behavioral RR(BRR), as indicated in figure 2.1. A BRR can be identified by two parameters Δ and w_0. Δ indicates the offset from the top of the screen and w_0 indicates the vertical height of the whole region. If a user wants to read the contents outside the RR, a scroll action, either scroll up or scroll down, is expected. In this research, we take the aforementioned assumption and model the user attention region as displayed in figure 2.1.

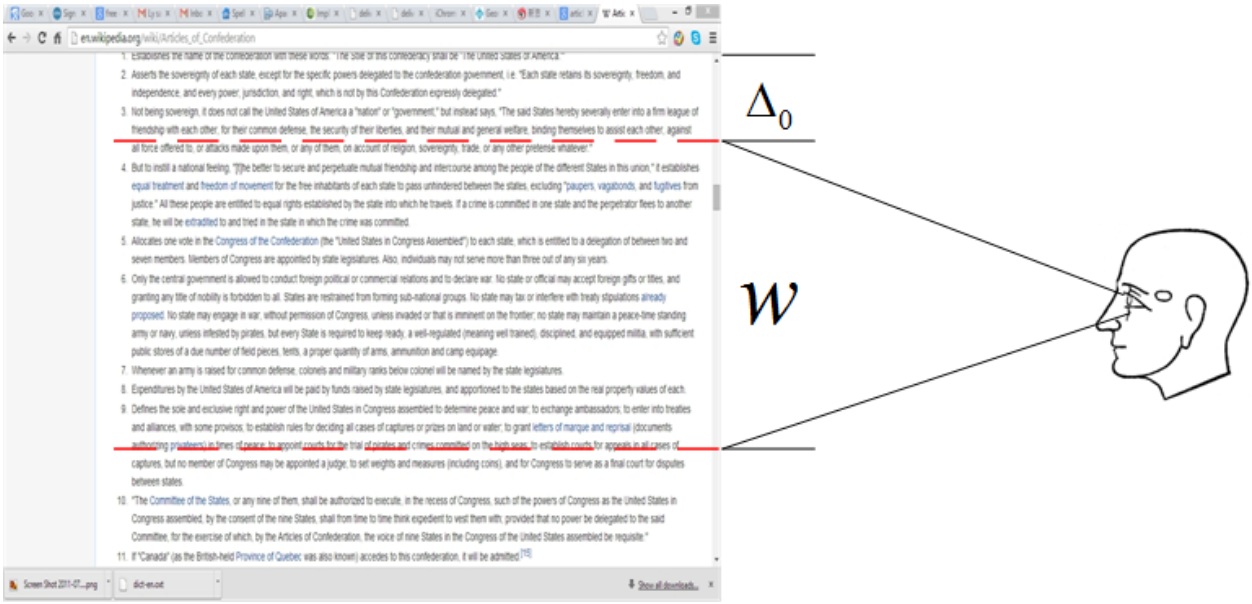


Figure (2.1) The preferred reading region on the screen

Taking this model into consideration, our next research question is: What behavioral data should be collected to compute RR?

Eye tracking data can be easily mentioned to answer the question, especially when a "visual" reading region is the goal. However, we argue that the eye tracking data has two major drawbacks: 1. The eye tracking collection requires extra hardware, which means the eye tracking based solution is not scalable and expensive. 2. Eye tracking is affected by the calibration problems, and movement limitations, etc. For these two reasons, we decided not to use eye tracking data. To meet the requirement of user-unaware and objective, we collect user interaction data during a user′s reading activity. In the next subsection, we will discuss how to collect and analyze user interaction data to get the user preferred region.

2.3.2 Calculating BRR by Interaction Data

When a user is doing daily reading, the keyboard input rarely happens during the reading process, we are more interested in the mouse-generated data. There are two categories of mouse-generated data: click and scroll. We consider the mouse-generated data to be informational behavioral data. To be specific, a mouse click is very likely to indicate the position of user′s instantaneous reading attention, while a page scroll indicates how much contents has been moved out of reading region.

In this subsection, we answer the question of how to infer BRR by mouse-generated data. To be specific, according to figure 2.1, RR can be identified by two parameters Δ and w_0. The computation of BRR can be transformed to how to compute Δ and w_0.

Now we examine a typical online reading activity, during which a user finishes reading a whole article. We assume the article is relatively long, which requires many scroll actions. The whole reading process can be illustrated by figure 2.2

In figure 2.2, a user performs a scroll action when he has finished reading contents of current reading region. At ith scroll action, a w_i of page height will be moved out of region. So when the article reading is finished, we have equation (2.1). In equation (2.1), Δ and w_0 are defined in figure 2.1, H is the overall height of the document. Ideally, equation (2.2)

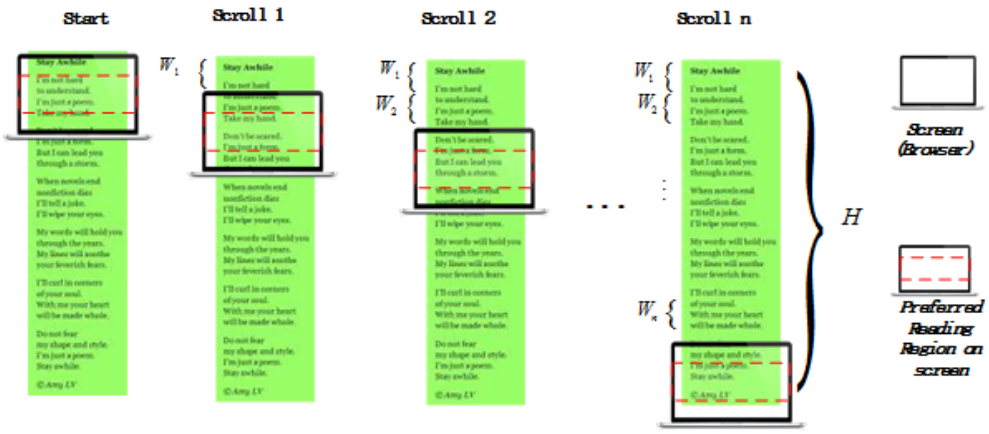


Figure (2.2) Reading of a long article in a browser

should hold and the parameter w_0 can be calculated in equation (2.3). In practice, since the scroll varies from time to time, it is unlikely for equation (2.2) to strictly hold, however, equation (2.3) can still be applied.

$$H = \sum_{i=1}^{n} w_i + \Delta + w_0 \tag{2.1}$$

$$w_1 = w_2 = ... = w_n = w_0 \tag{2.2}$$

$$w_0 = \frac{\sum_{i=1}^{n} w_i}{n} \tag{2.3}$$

With equation 2.3, we can compute w_0 parameter of a user's behavioral reading region(BRR). The computation of Δ can not be accomplished when only the scroll data is considered. Here we take the mouse click data into consideration. Since we take the assumptions that the reading process progresses line by line and the mouse click is an indication of instantaneous user attention, we take the first click after each scroll as the clue to compute Δ.

We note the click data K as a sequence $K = (K_0, K_1, ..., K_n)$, where there are n scrolls and the component K_i is the click sequence collected after i_{th} scroll. Furthermore, $K_i =$

$(K_i^0, K_i^1, ...)$ and K_i^j consists of timestamp and coordinates, $< t_{k_i^j}, x_{k_i^j}, y_{k_i^j} >$. So we can compute Δ as following equation:

$$\Delta = \frac{\sum_{i=0}^{n} y_{k_i^0}}{n+1} \tag{2.4}$$

Once we compute Δ and w_0, we can have a good estimation of BRR. BRR can be considered relatively stable. During the reading process in real time, BRR can be used to estimate the reading region if not enough information is provided. Due to the variation, equation (2.2) does not always hold, a realtime reading region(RRR) should be computed at each scroll action, especially when the variation among scrolls are relatively large.

2.3.3 Calculating RRR by Interaction Data

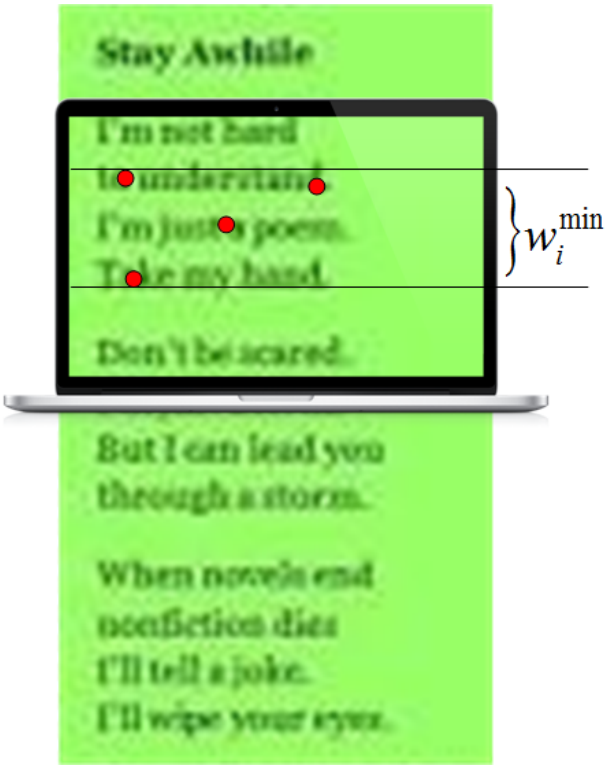


Figure (2.3) Mouse click and reading region

Since equation (2.2) does not always hold. If the variance of w_i is relatively large (it is

large in some cases), we need to consider the question of how to infer user reading region on each single scroll. Figure 2.3 displays the events happening during reading between two scroll actions. Here solid dots are collected mouse click data with the position on the corresponding coordinates on the screen. Since we assume that the reading progresses line by line and the mouse click is an indication of instantaneous user attention, we can calculate a minimum reading region after i_{th} scroll action, $w_i^{\min}$, which can be identified by the highest click and lowest click (as indicated in figure 2.3).

However, $w_i^{\min}$ is a minimum estimate of w_i. We can have a more accurate estimate if we take the neighboring scroll actions into consideration. In figure 2.4, we put the click data both before ith scroll (purple dots) and after $i+1$th (yellow dots). We can see there is a gap between two neighbor minimum approximates($g1$ and $g2$ in figure 2.4). Therefore, we can have a moderate estimate of w_i:

$$w_i = w_i^{\min} + (g_1 + g_2)/2 \tag{2.5}$$

It should be noted that, ideally, $w_i^{\min} = w_i = w_0 (1 \leq i \leq n)$ and $g_1 = g_2 = 0$. But in fact, it is unlikely that a user starts reading and finishes reading from the exact same locations. Click data has to be considered to compute RRR. If click data is not available, we use BRR to replace RRR.

2.4 What Have You Read and to What Extent?

Once we compute the BRR and RRR, we can answer our second research question: what and to what extents a user has read. The simple answer should be: the reader spent 6 minutes 45 seconds on this article. But this does not disclose much detail of the reading process. Here we want to answer this question with more details. Our goal is to compute user dwell time on each paragraph.

To formulate our solutions, we introduce the following notations. Given two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ on a screen at time instant t, the context identified by the two

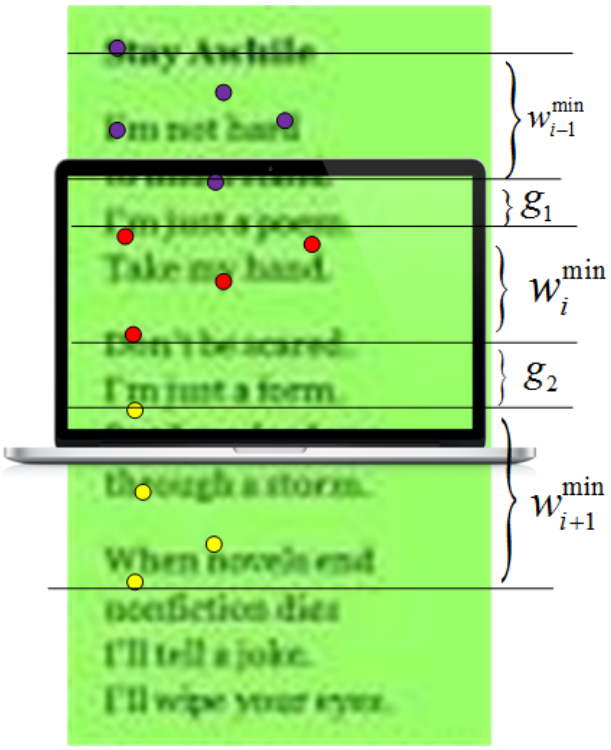


Figure (2.4) Mouse click and reading region in consecutive scrolls

points can be notated as : $C(p_1, p_2)$, which is indicated as dark background texts in figure 2.5. The corresponding time spent reading this context is $T(p_1, p_2)$.

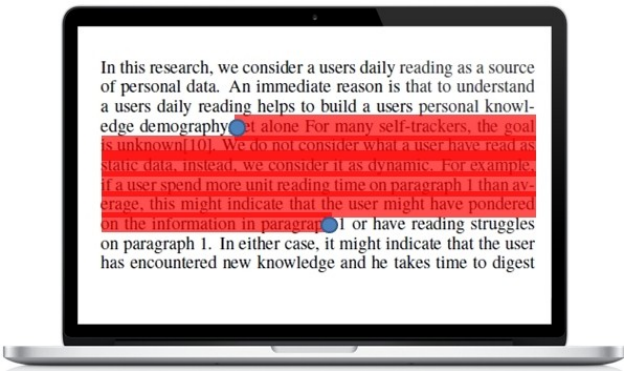


Figure (2.5) Contexts between points

For each scroll span w_i, p_s is defined as the left top point of that screen area, and p_e is defined as the right bottom point of that screen area. Intuitively, we can have the reading

time of this screen $T(w_i)$ as follows:

$$T(w_i) = T(p_s, p_e) \tag{2.6}$$

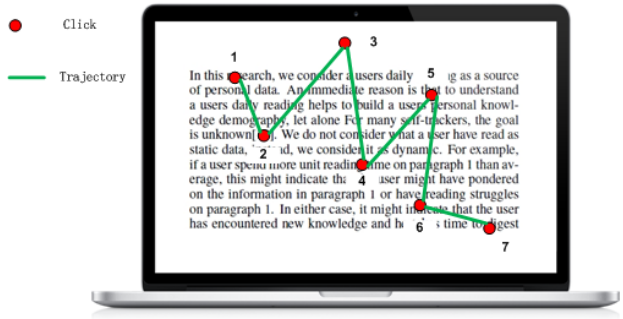


Figure (2.6) Mouse Click and Text Extraction

However, it is only a coarse-granularity estimation of the dwell time, because the context $C(p_1, p_2)$ is relatively large. By taking the mouse click data into consideration, we can have a fine-granularity estimation of dwell time. As we stated before, we consider mouse click as an indication of real time user attention. However, there might be some "noise" in mouse click data, which can not be considered a clue of user reading. Figure 2.6 illustrates a typical reading scenario where mouse click and mouse trajectory are displayed. Apparently, point/click p_3 is a "noise", since the position of p_3 does not indicate any text contents. This might indicate an occasional subconscious click. All the other six points locate in a text area. However, not all of these six points can be used as reading attention indication.

In real time, we note each click as $k(t, x, y)$, where t is timestamp, x, y are the coordinate of the point where click happened. The intentional click happens in a sequence of $k_1k_2k_3...k_n$. So, based on our assumption of line-by-line reading, we have the following equations hold, in equation 2.7, $1 \leq i < j < k \leq n$.

$$\begin{cases} C(k_i, k_j) \cap C(k_l, k_s) = \emptyset \\ t_i < t_j \leq t_l < t_k \end{cases} \tag{2.7}$$

According to equation 2.7, we can filter out p_5 (in figure 2.5) as noise since $C(k_1, k_2) \cap C(k_4, k_5) \neq \emptyset$.

In practice, the information of whether a click is intentional is not given, so we developed algorithm 1 to find the maximal intentional click set(ICS) where equation 2.7 holds.

Algorithm 1: Find the maximal ICS(intentional click set)

Input: The set of clicks $K = \{k_1, k_2, \ldots, k_n\}$
Output: The largest subset that meets equation 2.7
for $i \leftarrow 1$ ***to*** n **do**
  $S_i \leftarrow \{k_i\}$
for $i \leftarrow 1$ ***to*** n **do**
  if $S_i = \emptyset$ **then**
    continue;
  for $j \leftarrow 1$ ***to*** n **do**
    if $S_j \cup \{k_i\}$ *fulfills equation 2.7* **then**
      $S_j \leftarrow S_j \cup \{k_i\}$
      $S_i \leftarrow \emptyset$
$S_{max} \leftarrow ||S_1||$
for $i \leftarrow 1$ ***to*** n **do**
  if $||S_i|| > ||S_{max}||$ **then**
    $max \leftarrow i$
for k_i *not in* S_{max} **do**
  if $S_{max} \cup \{k_i\}$ *fulfills equation 2.7* **then**
    $S_{max} \leftarrow S_{max} \cup \{k_i\}$
return $S_{\max}$

After we apply the ICS algorithm on the collected data, we can get the dwell time of each paragraph. Since the text length(number of words) in each paragraph varies, we consider the unit reading time (ms/word) to be a more indicative measurement on dwell time.

2.5 Design and Implementation of UUAT

To verify our ideas, we designed and implemented an Unobtrusive User Attention Tracking(UUAT) system, which can accomplish two tasks after collecting user behavioral data:

1.UUAT calculates the user reading region, BRR or RRR. 2. UUAT infers what content(texts) and how long a user has read in an article. In this section we introduce our UUAT system and present our implementation details.

In order to analyze a user′s daily online reading activity, we prefer Browser-Client(BS) architecture to Client-Server(CS) architecture. The reason is that the BS architecture is more scalable and platform-independent. We choose browser plugin/extension as our implementation architecture, for the following considerations: 1. Compatibility. The plugin technology has been supported by all the mainstream web browsers. 2. Agility. The development cost and deployment cost of a plugin-based solution is low. 3. Privacy. All the collected data is under full control of a user himself. The user can keep all data as private and analyze it locally, or he can choose a cloud service to analyze his data anonymously.

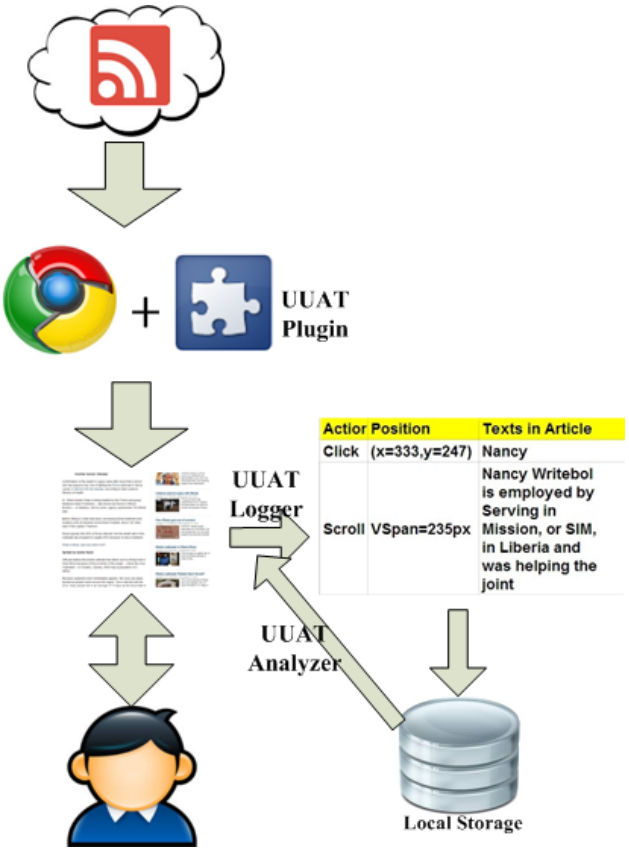


Figure (2.7) UUAT architecture

Figure 2.7 illustrates the architecture of UUAT as a Chrome plugin. Once user requested

an article from a website, the browser will first get the HTML contents for that article, then the html contents will be passed to UUAT and JavaScript code will be injected into the original HTML code. When a user is reading this article, his behavioral data will be captured and saved at a local position. In real time, user behavior can be analyzed with the help of historical data (if there is historical data) and reading context data can be collected.

UUAT collects three types of raw interaction data: MouseClick, Cursor and Scroll. The click and cursor are relatively easy to capture. By assigning handlers to *document.onclick* and *document.onmousemove* event listeners, UUAT can easily log the position where a click happens and the cursor trajectory. The log of user scroll is subtle. There are many cases in which a user scrolls up/down the screen more than once in a very short time interval, during which no actual reading happened. This happens very often when the user scrolls to a new position and aligns his reading region with the article contents before his scroll. To detect a real user scroll action, we adopted a backtracing window technology to rule out those adjustment-use scrolls,

2.5.1 Getting Click Data

Different readers have different reading habits. For many readers, mouse click interaction is a rare action during his daily reading activity. In this case, getting real click data become really hard. To get click data in this case, we developed a work-around technique.

As concluded by [25], cursor trajectory can be used as indication of use attention. In UUAT implementation, we combine cursor trajectory and the text on the webpage to render click data. As indicated in figure 2.8, UUAT turns each word in the article as a "motion sensor". When the cursor moves over that word. A javascript-triggered "click" will automatically be detected and logged.

2.5.2 Behavioral Data Analysis

The goals of behavioral data analysis are twofold: 1. Combined with historical data, it aims to calculate a preferred reading region so that in the future when less behavioral data

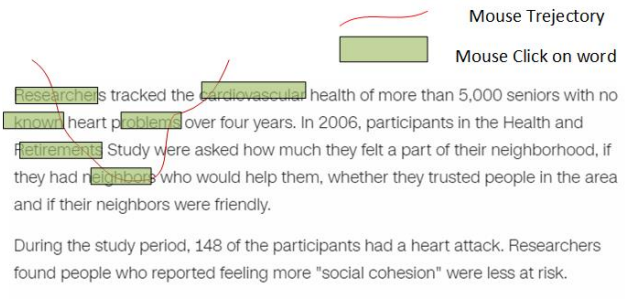


Figure (2.8) Getting clicks from mouse trajectory for non-click readers

is provided, an estimation of user reading can still be calculated by historical reading model. 2. For each reading activity, it calculates dwell time of each paragraph. We present our verification and experiment results of behavioral data analysis in next two sections.

2.6 Experiment and Verification

In this section we first present our design of experiments and verifications, then we present overall results to illustrate our two research goals: computation of RRs and computation of text dwell time.

2.6.1 Experiment Design and Procedure

The general idea of verification experiment is that we get verification data from a user′s reading in an unaware way. To make the experiments not affect a user′s daily reading, we told our subjects before the experiment that we would ask them to read an article in a browser. After they finished reading, we gave them several quiz questions based on the article they read, and a gift would be given if the quiz result was positive.

The design of the quiz questions was subtle. Through quiz questions, we hoped to identify quality subjects and help us build some ground truths of this subject′s reading process. We designed two types of questions in the quiz: article based question and self-evaluation question.

Article based question We divide the article into four parts. For each part we design a question. If a subject had read that corresponding part, he could answer the question

easily.

Self-evaluation question The ground truths we need are facts such as: which part of the article was difficult to read and in which part the read was relatively fast (skimming). So we designed four self-evaluation questions:

1. During your reading process, did you skip paragraphs (which means, you did not read those paragraphs at all)? If yes, can you please use "X" to tag those paragraphs in the given printed papers.

2. During your reading process, was there any paragraph that you read really fast(e.g. you scan the texts instead of reading them word by word)? If yes, can you estimate and underline those locations.(It does not have to be accurate, just estimate.)

3. Was there any paragraph that you read relatively slow? (e.g. having reading difficulties or reading one part of that paragraph/sentence multiple times). If yes, can you please tag those paragraphs on the printed pages with check ("✓")

4. Were you distracted from reading during the whole experiment, if yes can you please tag at which paragraph(s) you were distracted with (*)?If possible, can you estimate how many seconds you were distracted each time?

We printed out the article and asked the subject to indicate on the paper if necessary. It should be noted that the marks from the self-evaluation are not always accurate, since we do not know if the subject′s memory was accurate. However, this provides us part of the ground truth. Combined with other collected data (we will introduce them in the following part), we believe that we can construct an accurate ground truth for each subject.

The next question in the experiment design is: how to construct a ground truth? Since we can not completely depend on the answers from self-evaluation. , we setup a screencast to video tape the screen during the whole reading process. It should be noted that the screencast was hidden from the subject. By combining subject-transparent screencast video and self-evaluation answers, we can construct a solid ground truth.

We present the subjects an article from CNN news website, as indicated in figure 2.9. Although there are other elements on the webpage, such as links to other pages, commercial ads, in our research we do not consider the sides effects from those irrelevant elements.

Figure (2.9) Reading article

2.6.2 Evaluation of BRR calculation

We collected data from 35 subjects. By manually checking the papersheets, we eliminated two kinds of subjects: those subjects who failed to answer article-based questions and those subjects did not have enough mouse activities during the experiment. We consider the reading of those subjects to be not qualified for analysis. We ruled out 16 subjects and had 19 subjects left.

The first goal of our research is to identify the BRR and RRR of a user. According to equation 2.3, we compute the w_0 for each subject. The result is illustrated in table 3.1. In

table 3.1 we consider both "scroll forward" actions and "scroll back" actions. From table 3.1 we have two observations: 1. The variation of w_0 is relatively large. 2. Compared to the large variation of w_0, w_0 itself is only a very small portion of the whole window.

To find the the explanation of observation 1, we examined our data and we found out that there is a minimum size for the mouse scroll action. Each scroll action moves the whole page several integer of minimum size. For example, for subject with ID of 12, (part of) his scroll data is as follows: [53,53,-159,106,53,106,53,53,53,106,53,159,-53,106,106,53,-53,53,53,53,106,53,53,53] (unit is pixel and negative means scroll back). Here we can see that his preferred scroll distance is 53, as 29 out of 38 of his scroll action is 53(or -53). While if he wanted to adjust a little more space, the next size of scroll is 53*2. This minimum scroll size happened to all subjects who used mouse wheel buttons to perform scroll actions. On the other hand, we found that those data collected from those who used a laptop touch pad to perform scroll had smaller variations of w_0. For example, subjects with ID 16 and 13, carried out the experiments on a Macbook Pro, where the minimum scroll size is 1.

However, the minimum scroll size limitation can not explain the large variations, such as subject 2,4,14. We examined their data and found that this can be explained by their reading pattern. We will clarify this in the late user case analysis part(back and forth reader).

The observation 2 justifies our research. Generally, w_0 covers 10% to 20% vertical space of a window. It is important to locate which part of the screen draws the user attention. Only in this way can we estimate details of a reading activity.

The next evaluation of BRR computation is the accuracy of Δ. The value of Δ can be easily calculated according to equation 2.4. To verify the correctness of BRR calculation, we use the cursor heatmap as ground truth, which is illustrated in figure 2.10. In figure 2.10, we set up reading region as the vertical span where 68.2%($w \pm \sigma$) cursor trajectory points locate in. Table 2.2 listed the Δ calculation using heatmap as benchmark. From this table, we can see that the error of Δ varies from around 3% to 56%, but the compared to the window size(screen height), the error of Δ calculation is relatively small.

Figure 2.11 illustrates the corresponding locations of calculated BRRs on the windows.

Table (2.1) Calculation of w_0 of BRR(WH stands for Window Height)

ID	$\overline{w_0}(px)$	σ	$\sigma/\overline{w_0}$	$WH(px)$	$\frac{\overline{w_0}}{WH}$
1	73.05	37.68	51.60 %	635	11.50%
2	161.11	135.97	84.39 %	624	25.82%
3	96.19	43.36	45.08%	681	14.12%
4	195.73	190.91	97.53%	743	26.34%
5	123.08	42.13	34.23%	611	20.14%
6	55.79	11.83	21.21%	681	8.19 %
7	66.83	48.72	72.90%	635	10.52%
8	64.36	27.21	75.70%	681	9.45 %
9	65.89	25.86	39.25%	681	9.68%
10	72.27	28.64	39.63%	635	11.38%
11	153.85	63.43	41.23%	923	16.67%
12	65.47	25.89	39.55%	635	10.31%
13	122.22	41.57	34.02%	667	18.32%
14	115	187.81	163.32%	667	17.24%
15	100	0	0	624	16.03%
16	127.5	38.00	20.23%	624	20.43%
17	146.67	49.89	34.02%	624	23.50%
18	69.19	32.62	47.15%	681	10.16%
19	60.16	25.14	41.80%	681	8.83%

From this figure we can see that the reading regions of each individual differ from each other drastically: both the offset Δ or the vertical span w. This again motivates our research.

Since we do not have a benchmark for RRR calculation, we use the dwell time evaluation to verify the RRR calculation.

2.6.3 Evaluation of Dwelling Time

Once we compute the BRR and RRR, we are able to calculate the fine-granularity dwell time according to ICS algorithm. Since we do not have direct ground truth for dwell time calculation, we conduct several use case analysis in this subsection. We use reading speed(ms/word) to indicate the dwell time of a reader on a specific paragraph. Based on the reading patterns, we divide our subjects into three types: fluent reader, struggle/focused reader, back and forth reader. Among our collected subjects, we have 9 fluent readers, 7

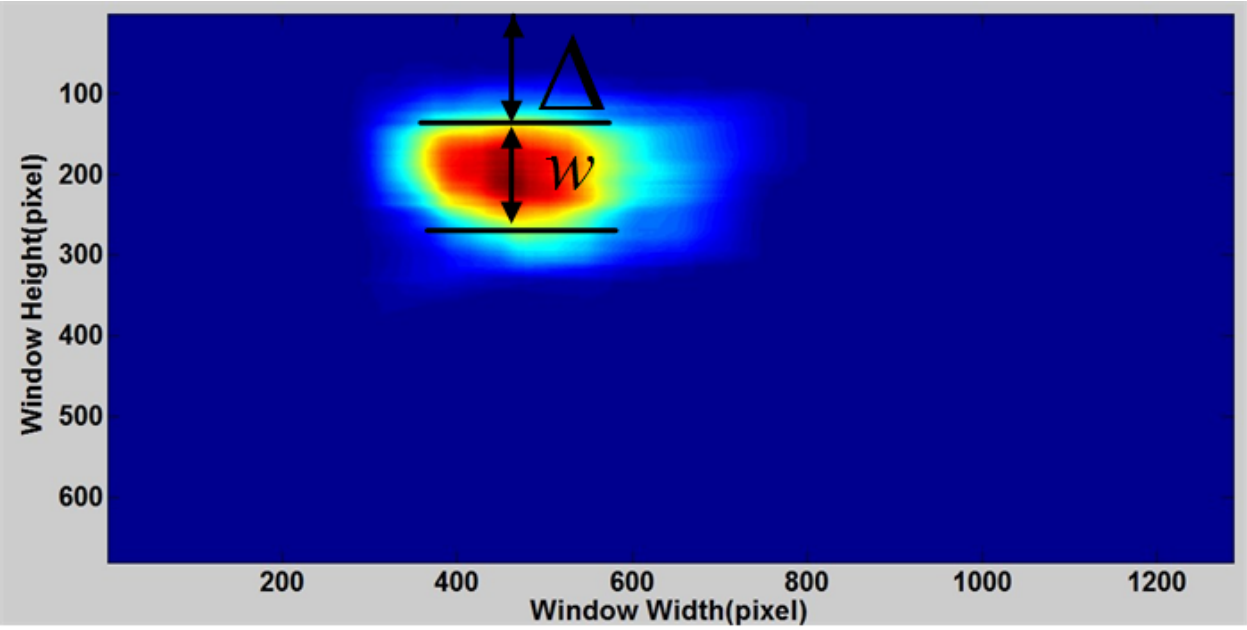


Figure (2.10) Cursor heatmap as ground truth

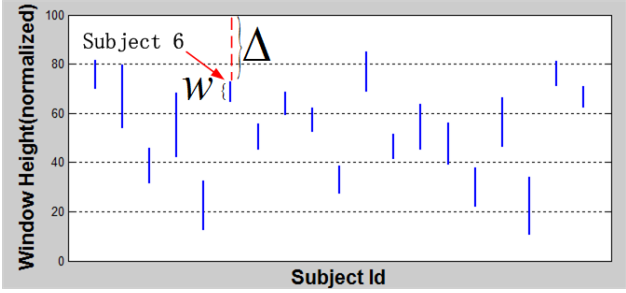


Figure (2.11) BRR distribution on window

struggled/focused readers and 3 back and forth readers.

Fluent reader We tag the first type of reader as Fluent Reader. For this type of reader, they read the articles paragraph by paragraph, no scroll back action happened during their reading process. Figure 2.12 illustrates the reading progression of a typical fluent reader(subject 11). In figure 2.12, the stairs curve indicates the scroll actions, the length of each horizontal part indicates the duration after that scroll action, while length of each vertical part indicates the vertical span on the documents. The continuous solid line in figure 2.12 indicates their real time word scanning process. Figure 2.13 illustrates the unit reading time (msec/word) on each paragraph, which is calculated partly by algorithm 1.

Table (2.2) Δ calculation using heatmap as benchmark

ID	Δ	$\Delta_{heatmap}$	Error	Error Rate	$\frac{Error}{WindowsHeight}$
1	156	118	38	24.36%	5.98%
2	289	127	162	56.06%	25.96%
3	333	370	37	11.11%	5.43%
4	241	235	6	2.49%	0.81%
5	368	413	45	12.23%	7.36%
6	145	186	41	28.28%	6.02%
7	205	281	76	37.07%	11.97%
8	241	213	28	11.62%	4.11%
9	212	258	46	21.70%	6.75%
10	350	389	39	11.14%	6.14%
11	155	137	18	11.61%	1.95%
12	270	307	37	13.70%	5.83%
13	289	243	46	15.92%	6.90%
14	247	292	45	18.22%	6.75%
15	338	387	49	14.50%	7.85%
16	160	209	49	30.63%	7.85%
17	368	411	43	11.68%	6.89%
18	193	128	65	33.68%	9.54%
19	140	198	58	41.43%	8.52%

For this type of "Fluent Reader", we can see that their reading speed is relatively stable at each paragraph. Although the BRR calculation of this indicates that the variation is not good (41.23%), but with our ICS algorithm, we can adjust the calculation of RRR at real time. We also investigated the answers of the self-evaluation questions of corresponding users. Their answers are "No", which means there was no special event (reading struggle, repeated reading or skimmed reading) happened during their reading experiments. The truths from the self-evaluation questions verify the correctness of our UUAT solution on this type of users.

Struggled/Focused reader We tag the second type of reader as Struggled/focused Reader. For this type of reader, they read the articles with occasional slower speed or back and forth, which we interpreted as either reading difficulties or focused reading. Figure 2.14

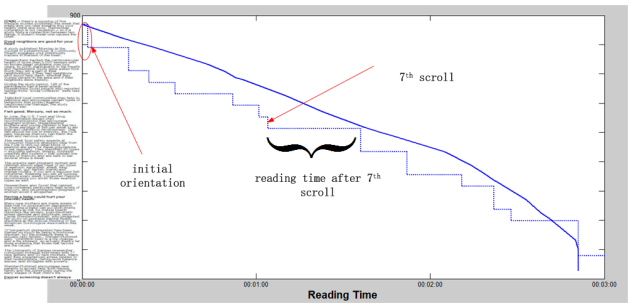


Figure (2.12) Reading Progression Map. Each user has an "Initial Orientation", during which he aligns his attention range and the head of given article. The stairs curve indicates the scroll action, horizontal part indicates time duration after each scroll, the vertical part indicates the span each scroll goes. The continuous curve indicates the words reading progression curve. Theoretically, the slope at each point indicates the instant reading speed.

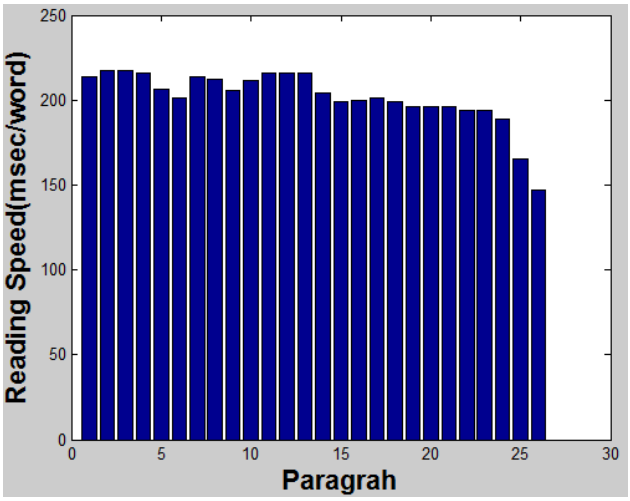


Figure (2.13) Reading speed on each paragraph(Fluent reader)

illustrates the reading progression map of a Struggled/Focused reader. From this figure we can see that the subject mainly read forward in a stable speed. But he had two repeated reading during the experiment. The "repeated reading" can be identified by a stair up curve(as indicated in figure 2.14). The repeated reading can also be illustrated from the reading speed diagram(figure 2.15). In figure 2.15, we can see that the reading speed at paragraph 13 through 15, 21 through 24 are relatively low. This partially matches the self-evaluation of this subject, which states that he read paragraph 15 and 22 twice or multiple times.

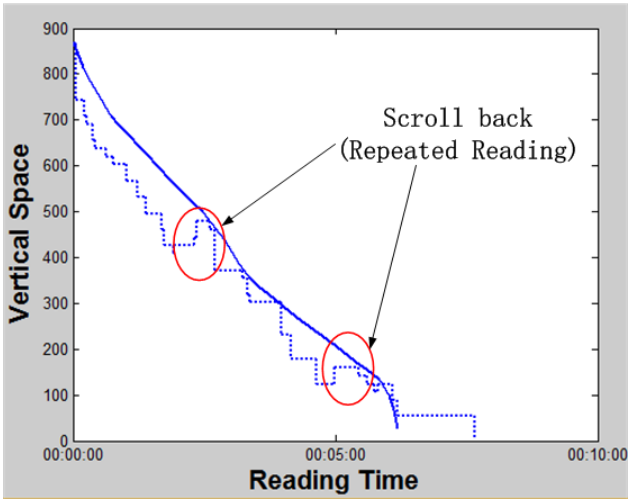


Figure (2.14) Reading Progression Map of Struggled/Focused reader

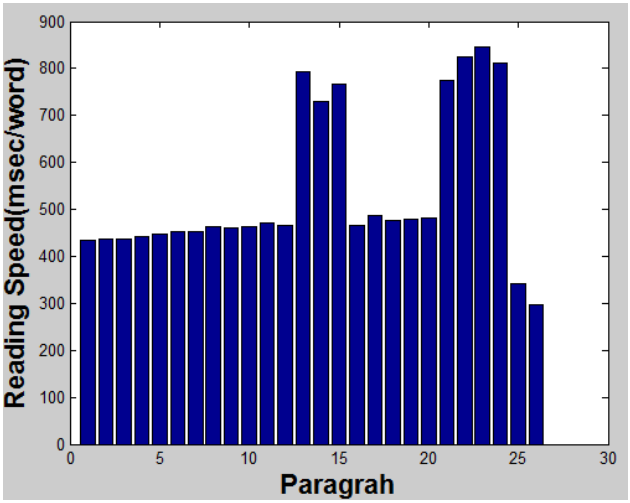


Figure (2.15) Reading speed on each paragraph(struggle/focused reading)

Back and forth We tag the third type of reader as "Back and forth reader". For this type of reader, they do scroll up and down very frequently. Figure 2.16 illustrates a typical scroll pattern of this user type. We can see that they do back and forth reading very frequently. In this case, UUAT does not work well since the attention range of the user varies a lot. We have subject 2, 4 and 14 as this type of reader, which partly explains why the computation of BRR is not accurate for them.

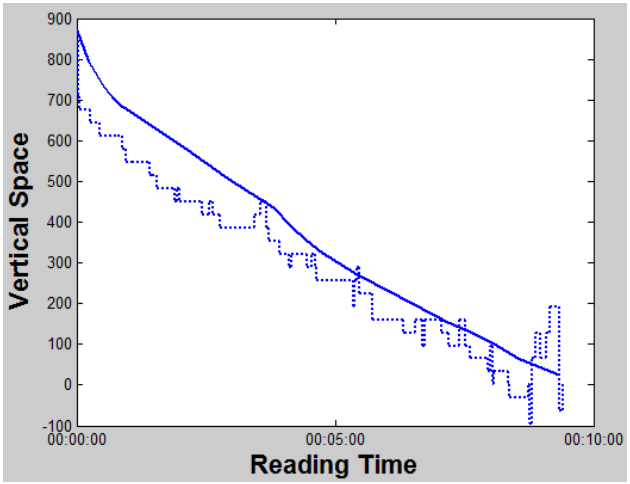


Figure (2.16) Frequent back and forth reading

Chapter 3

DATA VISUALIZATION AND MINING IN BIOLOGICAL DATA EXPLORATION

3.1 Introduction

Cancer biomarker research [37–39] is the process of identifying biomarkers with prognostic significance. A classic research method is identifying a logical stratification of the dataset so that one can stratify the whole dataset with one or more risk groups based on biomarkers, such as ER,PR,KI67. The stratified groups can be distinguished from others in terms of predefined outcome criterion(e.g death rate, prognosis, etc). This data analysis process can be complicated since a clinic dataset can be large and have many features/biomarkers/variables. The mainstream research paradigm is that the researchers import all the data to a spreadsheet. Then the researcher inspect the spreadsheet data "literally", that they read individual data record cell-by-cell and try to gain an overall insight of the dataset in their minds. After that, some common statistics methods, such as standard ANOVAs [40], may be applied so that insights of the whole dataset might be built. After an overall understanding of the given dataset is established, the researchers are able to carry out more discovery oriented approaches, such as survival analysis. Our work focuses on developing a data visualization platform to aid the preliminary dataset explorations stage and the following discovery process.

In current cancer biomarker research, data visualization is mainly utilized to present research results. It is underutilized, though, as a tool for early biomarker research, where it can assist in data exploration and pattern discovery. Previous work has shown that interactive visualization techniques can be useful tools for early dataset exploration. For example, PRISMA [8] and exploRase [9] can help researchers get a fast, intuitive, and comprehensive understanding of a new dataset. VizRank [10] helps find simple, interpretable data

projections that include only a small subset of genes yet do clearly differentiate among different cancer types. These solutions are either lack of interactivity or focusing only on the exploration stage.

To realize a more interactive and research-friendly platform, we have developed a visualization platform for cancer biomarker research that supports a broad range of data exploration and data analysis tasks. Our goal is to improve the efficiency of data exploration and analysis mining(pattern discovery) in cancer biomarker research.

Currently, a researcher conducts preliminary data exploration by directly inspecting the data in a spreadsheet. This method is not efficient for high dimensional data sets because the connections between multiple data dimensions are often implicit in a spreadsheet or a single-view plot. Our visualization platform provides the following features to help improve this exploration process.

- **Multiple views:** This is a useful visualization technique for exploring high dimensional data. A given dataset is considered to be a conceptual entity. Each view allows the user to investigate one aspect of the conceptual entity either by displaying a subset of the data or a subset of the data dimensions.

- **Synchronized views:** User interactions are fully synchronized in multiple views. For example, if a data entry is selected in one view, all the corresponding data points are automatically selected in other views. Synchronized views are important for high dimensional datasets since it keeps all the presented views coherent and facilitates the discovery of non-trivial data relationships [41].

- **Extensible and customizable:** Our visualization platform is extensible and scalable because it is designed as a web service. Users can quickly add or remove charts to create different combination of multiple views. We make the dataset format to be Comma Separate Values(CSV) since almost all database supports CSV conversion.

- **Visualization-aided research technologies:** Our visualization platform not only supports preliminary exploration, but also supports common data analysis tasks in

survival analysis, such as our a data analysis module, namely CutPointVis, which enables a quick, convenient and intuitive cut point determination.

We developed CancerVis, an interactive explanatory platform for cancer biomarker research. CancerVis helps a researcher explore high-dimensional datasets with techniques such as scatter plot, parallel coordinates and color coding. CancerVis is extensible and scalable because it is designed as a web service. In CancerVis, a dataset can be visualized from different perspectives, according to the user customization. A new chart can be added to the client′s viewport and any existing chart can be removed from current viewport. Therefore, this allows only useful plots to be kept in the user′s viewport. CancerVis automatically detects data ranges and data types for its given dataset. Visualization is synchronized among all plots in the viewport: selection of entries in one plot result in highlights of corresponding entries in the remaining plots. Furthermore, CancerVis tackles survival analysis by providing CutPointVis, an interactive visual tool for cut point determination in survival analysis of cancer research.

3.2 Related work

3.2.1 Interactive Data Visualization Platforms in Cancer Biomarker Research

According to [42], "data visualization transforms literal data into intuitive charts and enables the easy interpretation of a dataset". The objective of a data visualization platform is to ease data understanding, exploration, so that underlying patterns are easily uncovered [43]. To assist scientific data analysis, information/data visualization tools allow users to manipulate dataset (e.g data filter, adding/removing individual interesting data items) and visually rearrange high-dimensional data. Common features of data visualization include the following.

- Overview of a given dataset
- Filtering: certain queries can be input into system to rule out unnecessary items.

- Selection on demand: individual items can be added/removed from current view.
- Zooming: both zoom in and zoom out should be supported when exploring a plot.

In biomarker research, there is not a comprehensive visual data exploratory tool. Moreover, very few existing tools provide visualization support for data mining in biomarker research.

[**?**,44–47] used visualization tools in biomarker research. But they only used data visualization to present results, not for data exploration. [8,9] are classic work on data visualization in cancer research. Explorase [9] is a coordinated multiple-view system that presents a logical data set with multiple visualization technologies. Although it is useful for the exploration of biology data, it lacks interaction techniques. Prisma [8] is an interactive multiple-view visualization for data exploration. Our platform has not only similar visualization functions as Prisma but also visualization tools for data mining.

Data mining related visualization techniques have not been widely applied in cancer research. The researchers of [48] presented a visualization web service that helps find the cutpoint of a given dataset. It provides up to five different implementations of cutpoint determination algorithms. Our platform distinguishes itself from the previous one by providing better interactivity. All the visualizations generated by [48] are static while our platform provides interactive visualizations. With our platform, users can quickly visualize the optimal cutpoint determinations. Furthermore, comparisons between different cutpoints can be visualized so that a more reasonable cutpoint can be chosen.

X-tile [49] provides a free tool to analyze survival time. Our platform provides better user interactions and additional cut-through analysis tool. Moreover, our platform uses a different log rank statistic method, which is more scientifically reasonable, as the optimization algorithm.

3.2.2 Interactive Visualization Tools for Cancer Biomarker Data Analysis

As a pioneer work in cutpoint visualization, X-Tile [49] focuses on dividing a dataset into three subsets. X-Tile precomputes the division results at each possible pair of cutpoints. Af-

ter that, it populates a 2-D right-triangular grid for a researcher to choose a pair of cutpoints. At each cutpoint pair, the researcher can visualize the corresponding Kaplan-Meier [50] plot (K-M plot) and histogram analysis. Unlike X-Tile, CutPointVis focuses on dividing a dataset into two subsets. Furthermore, CutPointVis provides realtime interactivity: it does not pre-compute all the result before any user interaction, since the pre-computation may take too much resource or time, especially when a large dataset is given.

Cutoff Finder [48] provides static visualization of potential optimal dichotomization cutpoints. After loading a dataset, it computes an optimal cutpoint according to a user selected optimization method. Although CutOff Finder optimizes the Cox Regression groups p-value, it does not allow flexibility or group stratification exploration. Therefore, with CutOff Finder there is no way to identify other potential suboptimal cutpoints (which can be only slightly inferior but lay on very different parts of the continuous biomarker spectrum) or identify stratification patterns (such as the high risk group survival with increasing threshold selection).

Besides aforementioned work, there is few research in visualization tool for cutpoint optimization. We solve this problem by developing CutPointVis. Besides providing a fast and convenient graphical tool for biomarker optimization, CutPointVis offers interactivity for researchers to explore other context-dependent optimal cutpoints.

3.3 Overview of CancerVis

CancerVis is a platform for cancer biomarker research. It provides interactive data exploration tools and scientific research tools. The design guidelines are as follows:

- **Data Compatibility** CancerVis uses CSV files as standard data input, since almost all data format can be converted into CSV format. Considering that CSV format does not provide basic data information, such as property types and value range, CancerVis conducts a data pre-processing to restore these information.

- **User Friendly GUI** CancerVis provides easy-to-use GUIs for each interactive explo-

ration task. Interacting with data can help a researcher quickly gain insight into the dataset.

- **Multi-view Visualization** Synchronized multi-view visualization is essential to CancerVis. The synchronized interactions across different views help a user examine the data from different perspectives.

- **Implementation of visualization techniques** CancerVis implements two basic data visualization techniques: scatter plot and parallel coordinates plots. Both plots can be customized by users based on the input data.

- **Data mining tools** CancerVis provides CutPointVis, an data analysis tool that can expedite finding optimal cut point in survival analysis.

Figure 3.1 shows the main working space of CancerVis.

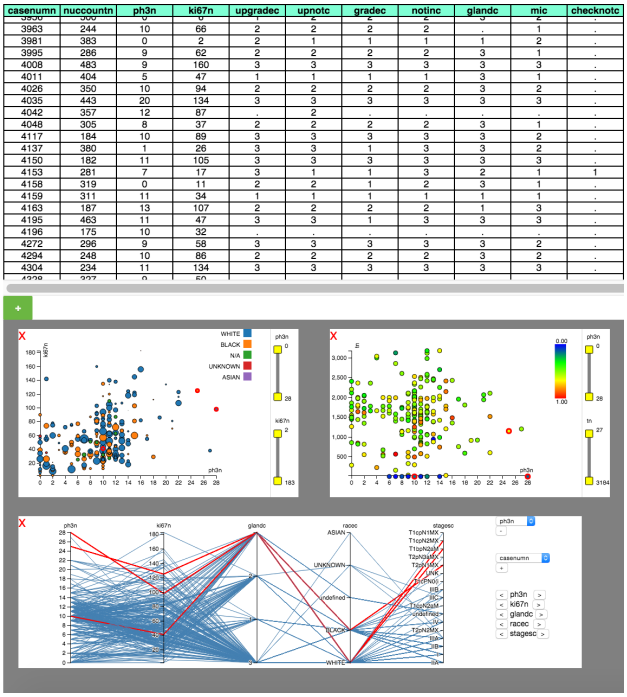


Figure (3.1) CancerVis overview

In the following sections, we discuss data exploratory functions and data analytic tools of CancerVis.

3.4 Exploratory Visualization Functions

The data visualization functions are designed for a researcher to explore different aspects of a given dataset. Users can view data in a table and one or more scatter plots or parallel coordinates. Interactions with one view is automatically synchronized in other views. For example, when a data item is selected in one view, the same data item are highlighted in other views.

3.4.1 Literal Dataset

CancerVis uses CSV as its data input format. CSV format is easy to parse and almost all mainstream database software support CSV output. The drawback of CSV input is that it loses data type information. To solve this problem, CancerVis uses a data pre-processing step to retrieve data types from a CSV input file. Figure 3.2 indicates the pre-processing flowchart of CancerVis. Currently, CancerVis recognizes five data types: Date,String,Decimal,Integer and Categorical. In a Categorical type, numbers are treated as strings.

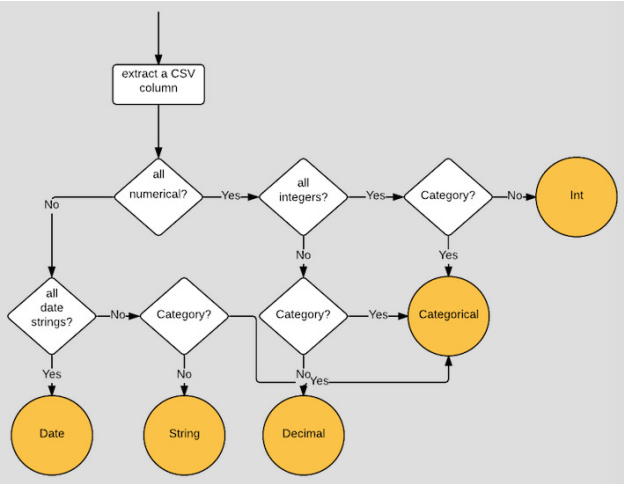


Figure (3.2) CancerVis Data Preprocessing

After data pre-processing, CancerVis presents the dataset in a table, namely VisSheet. VisSheet allows users to check the original data entry at any time. So VisSheet is designed as a permanent GUI component and cannot be closed.

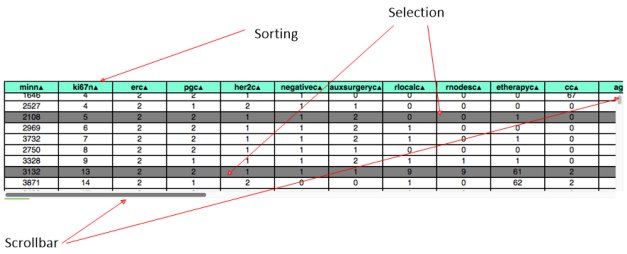


Figure (3.3) VisSheet

Figure 3.3 shows the GUI of VisSheet. Large tables can be viewed with horizontal and vertical sliding bars. VisSheet is different from regular HTML based tables because it has two features specially designed for biological research.

- **Sorting:** Researchers often need to study data in a certain order. VisSheet provides sorting button on each cell of head line. Currently VisSheet only sorts numerical fields, as indicated in figure 3.3.
- **Synchronized Selection:** As indicated by figure 3.3, users can select one or more data items in VisSheet and the same item will be highlighted in other views. This feature will be further discussed later.

3.4.2 Scatter plot

Scatter plot is a powerful data exploration tool [51]. CancerVis′ scatter plot module is named as VisScatter. VisScatter can visualize four dimensions using two axes, point size, and point color.

When a scatter plot is created, a user selects data columns for the X and Y axes. Optionally, the user can select a third dimension for point size and a fourth dimension for color coding, as indicated by figure3.4. The dimensions selected for X,Y and point sizes must be numerical(integer or floating point) and the dimension selected for color coding must be categorical.

Selection Users can select data points in VisScatter through double-clicking. The selected data entry will be highlighted with a red circle, as indicated in figure 3.5.

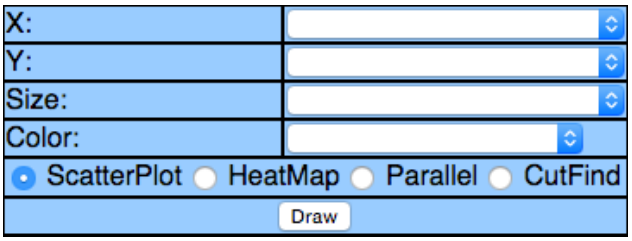


Figure (3.4) Dimensions in VisScatter

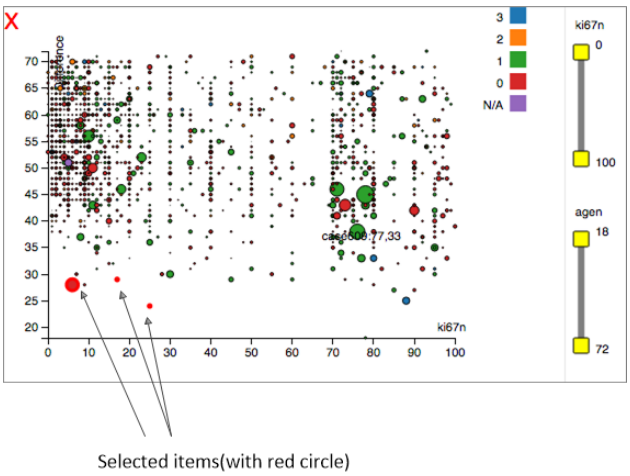


Figure (3.5) Data selection in VisScatter

Filters In VisScatter, users can filter out certain data items through a GUI interface. Since both X and Y axes are numerical properties, two sliding bars are added to the right of the plane, as indicated by 3.6. While the fourth dimension is categorical, legend bars are added to the planes, as indicated by figure 3.7.

3.4.3 Parallel Coordinates Plotting(PCP)

Parallel coordinate [52] plot is a useful tool for visualizing high dimensional datasets. CancerVis provides a parallel coordinate tool called PCP (figure 3.8). CancerVis provides user interactions such as selection and brushing (figure 3.8). Although PCP can visualize data of any dimension, the screen size limits the number of dimensions that are visible at the same time. Therefore, in most cases only a subset of the dimensions are presented within a PCP. However, in PCP, a researcher can quickly add dimension, remove dimension, or rearrange the order of the dimensions. In figure 3.8, the buttons in green dotted rectangle *a*

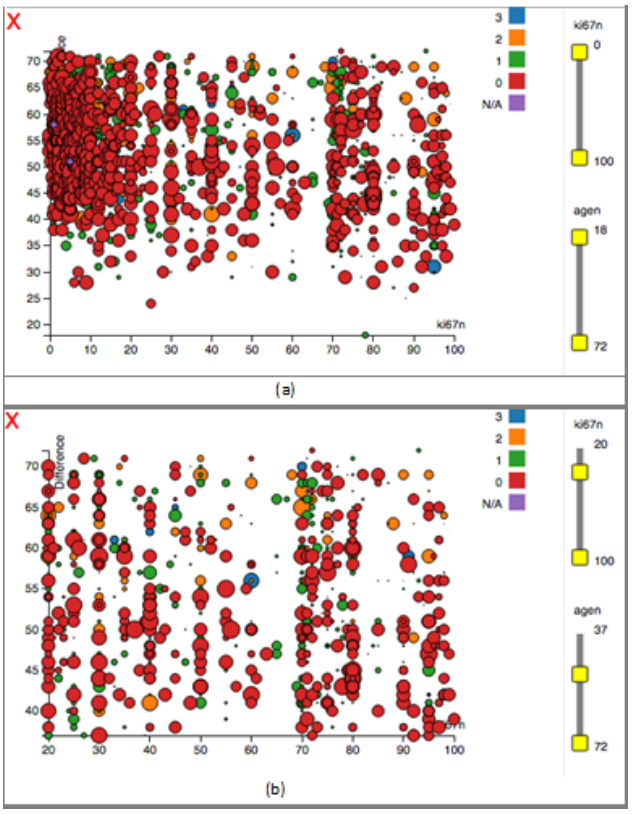


Figure (3.6) Numerical filters in scatter plot.(a) Before filtering (b) both X and Y axes are filtered

serve for adding and removing axes while the buttons in green dotted rectangle *b* serve for rearrange order of axes.

3.4.4 Inter-view Synchronization

In previous multi-view exploration systems, such as [9], the views are not synchronized. We believe that synchronization of user interactions across all views is important for analyzing high dimensional data. Therefore, CancerVis implements a synchronization mechanism that allows any interaction in one view to be synchronized with all other views. Figure 3.9 shows the synchronization mechanism of CancerVis. Since any interaction (selection in figure 3.9) in a view involves one or more data entries, the same interaction is applied to all the counterparts in other views.

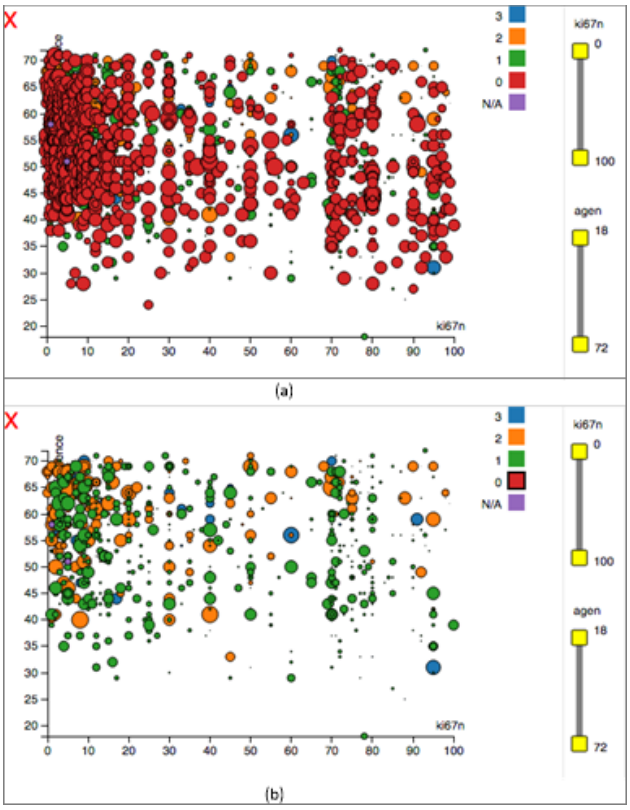


Figure (3.7) Categorical Filters in VisScatter.(a) Before filtering (b) After value "1" (red circles) is filtered

3.5 Mining Cancer Data with CutPointVis

Besides the aforementioned pure visualization features, CancerVis provides an interactive module, CutPointVis, which supports common data mining techniques in survival analysis for cancer biomarker research, such as visual assistance for determination of the optimal cutpoint. CutPointVis provides a full-fledged visual exploration support for data mining in survival analysis, which includes: 1) Realtime plot of Kalpan-Merier curve based on user selected cutpoints. 2) Displaying the Log-Rank statistics when stratifying for outcomes for multiple categorical classes (i.e. grade, stage, or lymph node status).

In the following parts of this section, we first briefly introduce the principle of cutpoint optimization in survival analysis(Log Rank Statistics), then we present the two aforementioned interaction features.

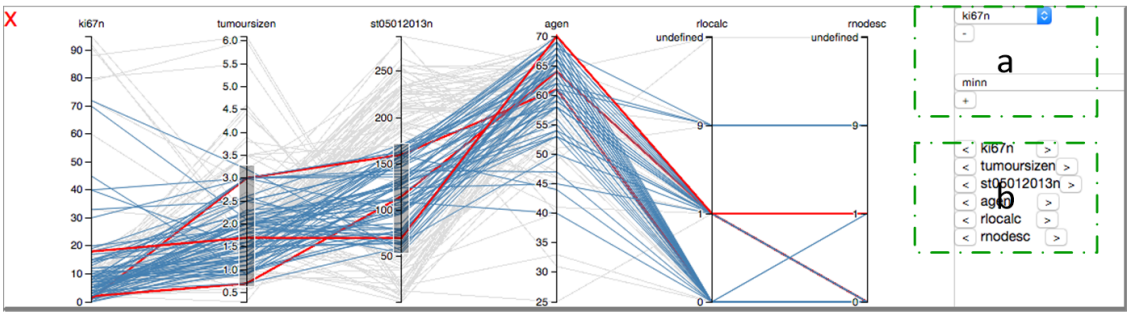


Figure (3.8) Parallel Coordinate Plot (Three selections and two brushings on PCP, axis rearrangement and adding/removing axes)

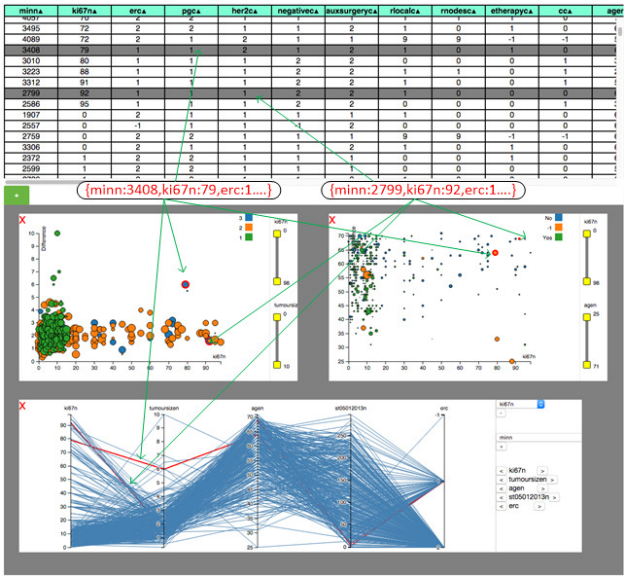


Figure (3.9) CancerVis synchronized interactions

3.5.1 Cox Model for Optimal Cutpoint Determination

During cancer research, when a survival analysis is conducted, it is a common practice to dichotomizing a continuous covariate [53]. Compared to data dependent methods, a log rank test statistic based method [54] finds a statistically optimal solution [55]. However, due to relatively high computational cost, log rank test statistic based cutoff point determination method [54] has not been widely used until recently.

To conduct cutpoint determination on a given dataset G, a user first chooses the risk factor R and the time-to-event variable T. R is a continuous variable and T is the time until an event happens. We use $t_{(1)} < t_{(2)} < ... < t_{(k)}$ to denote the ordered observed events of

outcome variable T. We use C to denote set of K distinct values of the continuous variable R. Given a hypothetical point P in G where $R = p$, it divides the dataset into two groups, group with $R > p$ and group with $R <= p$. Let $d_{(i)}$ denote the number of events at time $t_{(i)}$, $r_{(i)}$ be the number of subjects at risk prior to time $t_{(i)}$. Let $d^+_{(i)}$ denote the number of events at time $t_{(i)}$ in group $R > p$ and $r^+_{(i)}$ be the number of subjects at risk prior to time $t_{(i)}$ in group $R > p$. The log rank test statistic(LRS) of cutpoint p can be computed as in [54]:

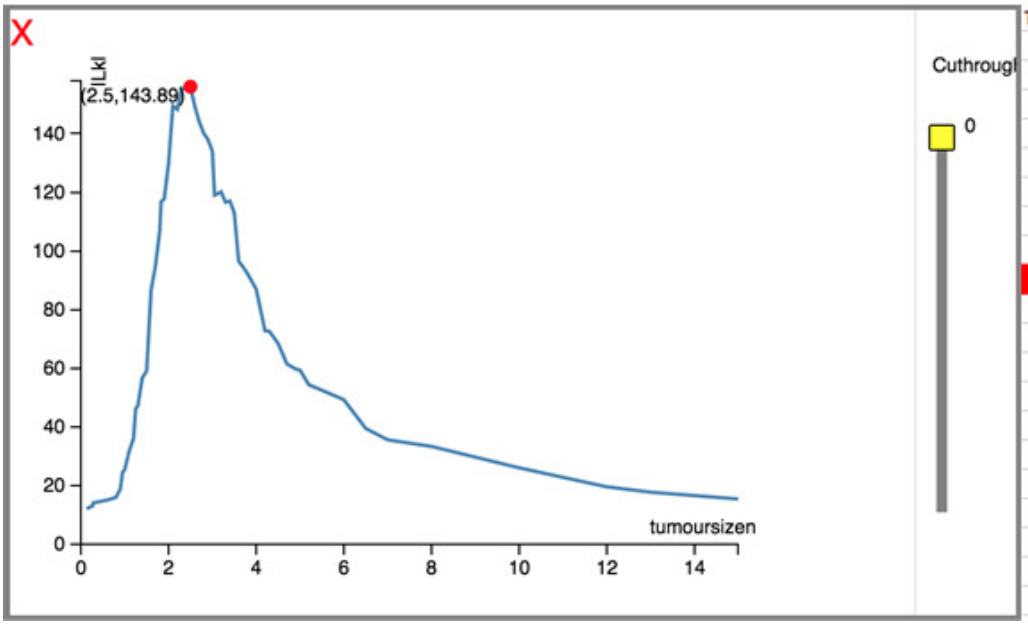
$$LRS(p) = \sum_{i=1}^{K} (d_i^+ - d_i \frac{r_i^+}{r_i}) \tag{3.1}$$

The optimal cutpoint P_m is the point that maximizes the absolute value of LRS. Statistically, a stratification at $R = P_m$ maximizes the difference between two groups. This procedure is usually carried out by a scientific computation platform, such as SAS or Matlab. Although the optimal point P_m can be determined by manually reading literal LRS results or by an excel/SAS max function, it is not efficient or intuitive, as indicated in the spreadsheet on the right part of figure 3.10 (The left part is the LRS curve plotting provided by CutPointVis). Furthermore, as indicated in figure 3.11, besides theoretically optimal point, there are other suboptimal points that are worth investigating (black dots on the curve). These suboptimal points are not likely to stand out by human inspection on a simple text/sheet output.

3.5.2 Realtime K-M Plotting for Cutpoint Determination

According to Cox model, the optimal cutpoint p_m is the point that maximizes the absolute value of LRS. However, as indicated in figure 3.11, besides the theoretically optimal cutpoint, there are other suboptimal cutpoints that are worth investigating(black dots on the curve). Furthermore, as stated in [56], "There is no single method or criterion to specify which criterion is best and thus the results of analysis from different categorization methods may be different".

To satisfy different criteria for cutpoint optimization practice in survival analysis, we developed CutPointVis, an interactive exploration tool for cutpoint analysis.

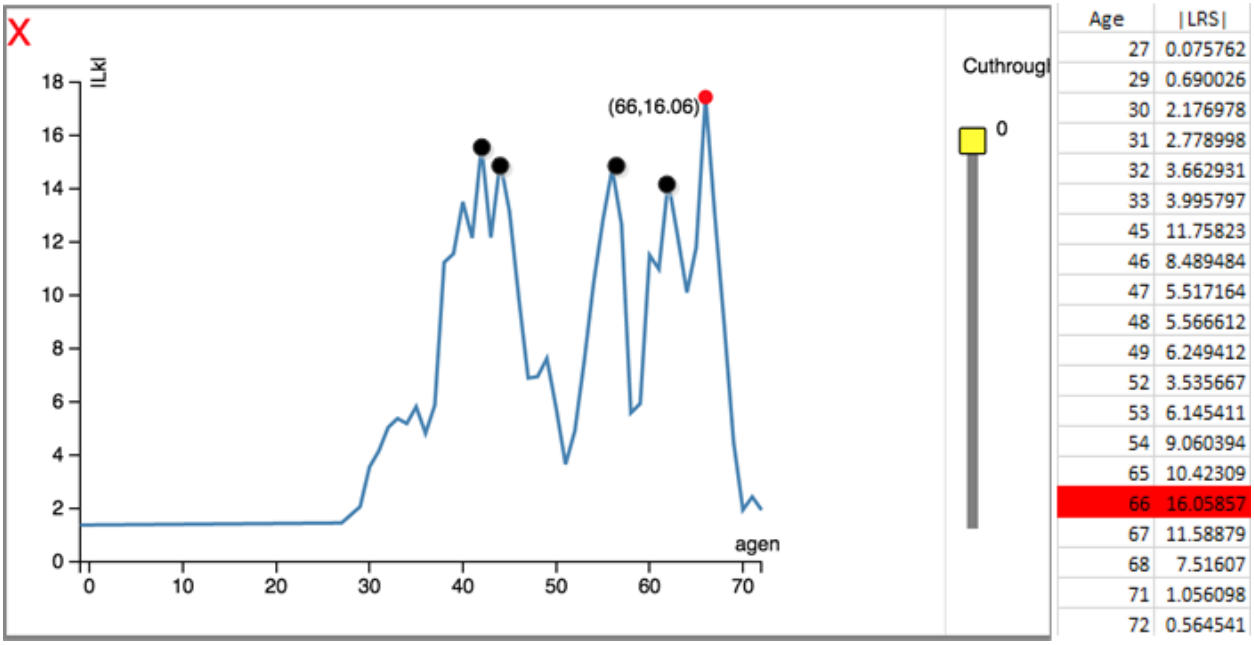


TumorSize	|LRS|
0.13	0
0.27	0.994634474
0.28	1.991944474
2	117.750126
2.1	137.3897543
2.2	136.1806581
2.3	143.4282148
2.4	142.8113214
2.5	143.8885482
2.6	137.1372224
2.7	131.9473847
3.7	82.42370714
3.8	80.17621024
4	74.85079126
4.2	60.67945275
4.7	49.36002244
4.9	47.63785907
12	7.474007916
13	5.649338611
15	3.324669305

Figure (3.10) Determining optimal LRS point by a figure(left) and by a spreadsheet(right)

Figure 3.12 illustrates the visualization interface of CutPointVis. In this visualization, a researcher is able to select risk factor R, outcome variable T and censoring indicator. Then a LRS curve can be plotted with the optimal cutpoint highlighted, as indicated in figure 3.13. From figure 3.13 we can see that the Cox optimal cutpoint is indicated with a red circle at $R = 11$.

Besides presenting optimal cutpoints for one group, CutPointVis provides a function to visualize optimal cutpoints for separate groups. A dataset can be split by a "Grouping" factor(as in Figure 3.12). Then the optimization can be applied to all the categories in the group, as indicated by figure 3.14. This plot represents the change in log likelihood when comparing patients above versus below the selected thresholds of the x-axis variable. The color of the lines indicate the grade (aka group) of the patients and the red dots show the value where optimal survival stratification is observed. We can interpret this data as clinical evidence in support of variable grade determined thresholds for cellular proliferative risks, with grade 3 patients showing highest risk in maximal Ki-67% while grade 1 and 2 show much better survival with minimal Ki-67%. We can also see that the ideal thresholds for grade 1 and 3 are not due to noise, and instead indicate real maximal survival stratification



Age	|LRS|
27	0.075762
29	0.690026
30	2.176978
31	2.778998
32	3.662931
33	3.995797
45	11.75823
46	8.489484
47	5.517164
48	5.566612
49	6.249412
52	3.535667
53	6.145411
54	9.060394
65	10.42309
66	16.05857
67	11.58879
68	7.51607
71	1.056098
72	0.564541

Figure (3.11) One optimal point and four sub-optimal points

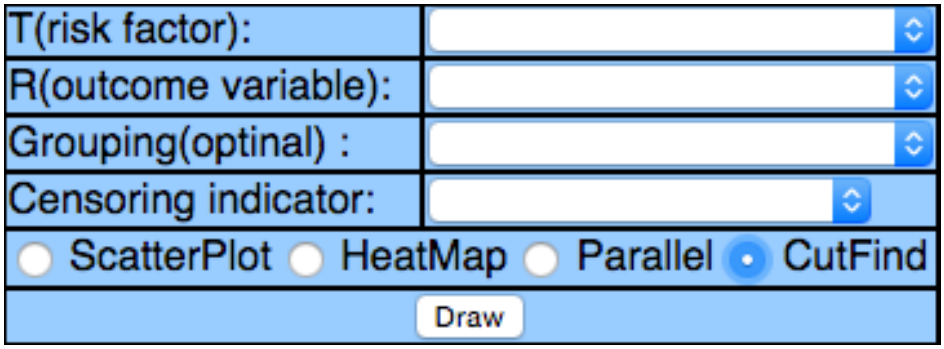


Figure (3.12) Select variables for cutoff point analysis

whereas for grade 2 the ideal threshold is a bit ambiguous because there exists multiple peaks at multiple Ki-67 values which show a similar log rank statistic. Also, using this type of data easily allows a biostatistician to identify if any threshold can be used regardless of grade to allow for suitable overall stratification (such as around 10 for the 3 grades).

As we stated before, due to different research contexts, Cox optimal cutpoint may not be an expected optimal cutpoint. To handle this flexibility, CutPointVis provides a feature for a researcher to further visually explore other potential optimal cutpoints.

A researcher may choose a different optimal cutpoint(other than Cox optimal cutpoint)

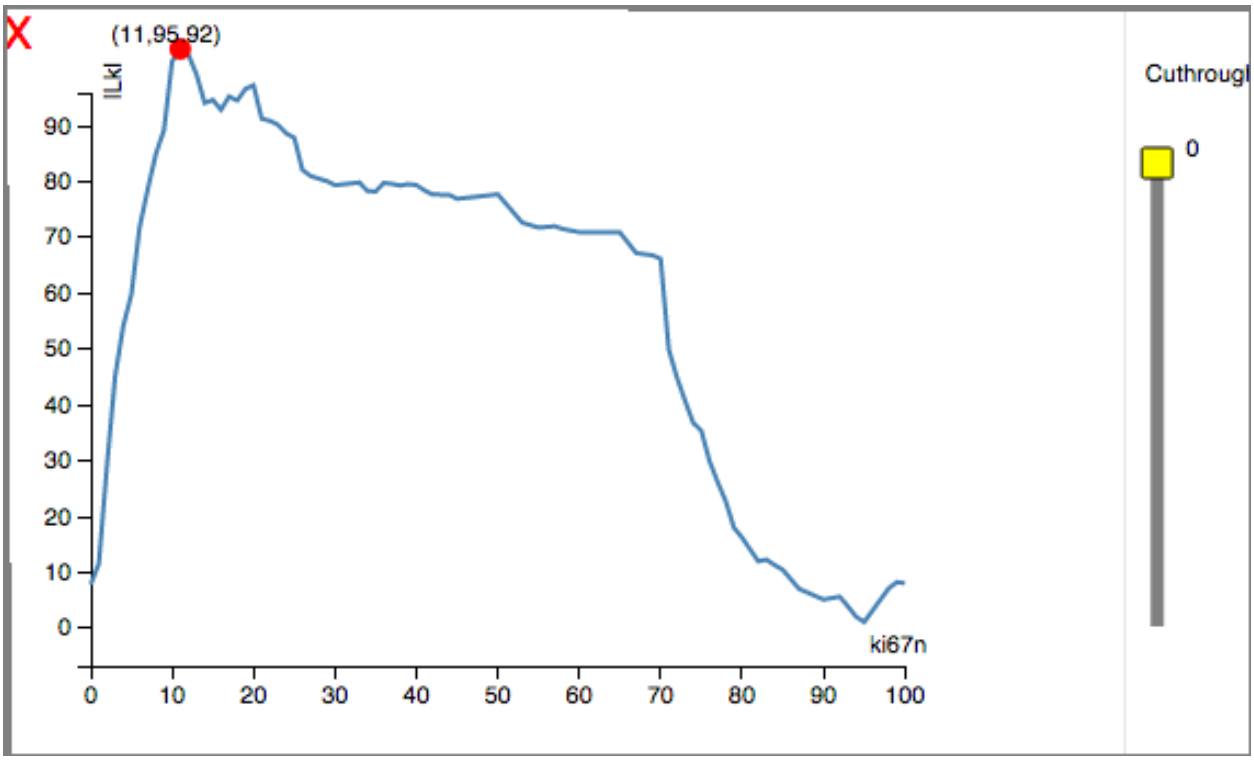


Figure (3.13) Visualizing optimal LRS cutpoint

due to many reasons, but in most of the cases a Kaplan-Meier estimator [51] would help the researcher to determine a context-dependent optimal cutpoint. Observing this fact, CutPointVis provides an interactive Kaplan-Meier plotting feature.

After a researcher has chosen variables R and T for survival analysis as in figure 3.12, a LRS plot will be presented as in figure 3.13 and the Cox optimal cutpoint (global maximum point) is tagged with a black dot. After that, as indicated by figure3.15, the researcher can explore different cutpoints with the cursor.

In figure 3.15, a user operates and locates the cursor to any point of a LRS curve. Meanwhile, the vertical sweep line of cursor determines a cutpoint P_c which is the intersection of the sweep line and LRS curve. Using p_c as cutpoint, a K-M plot will be presented over the LRS curve, the cutpoint is also labeled. In this way, a researcher can explore every single cutpoint, and visualize the corresponding K-M plot effect at the same time. We believe that realtime K-M plot offers researchers more flexibility and convenience to explore

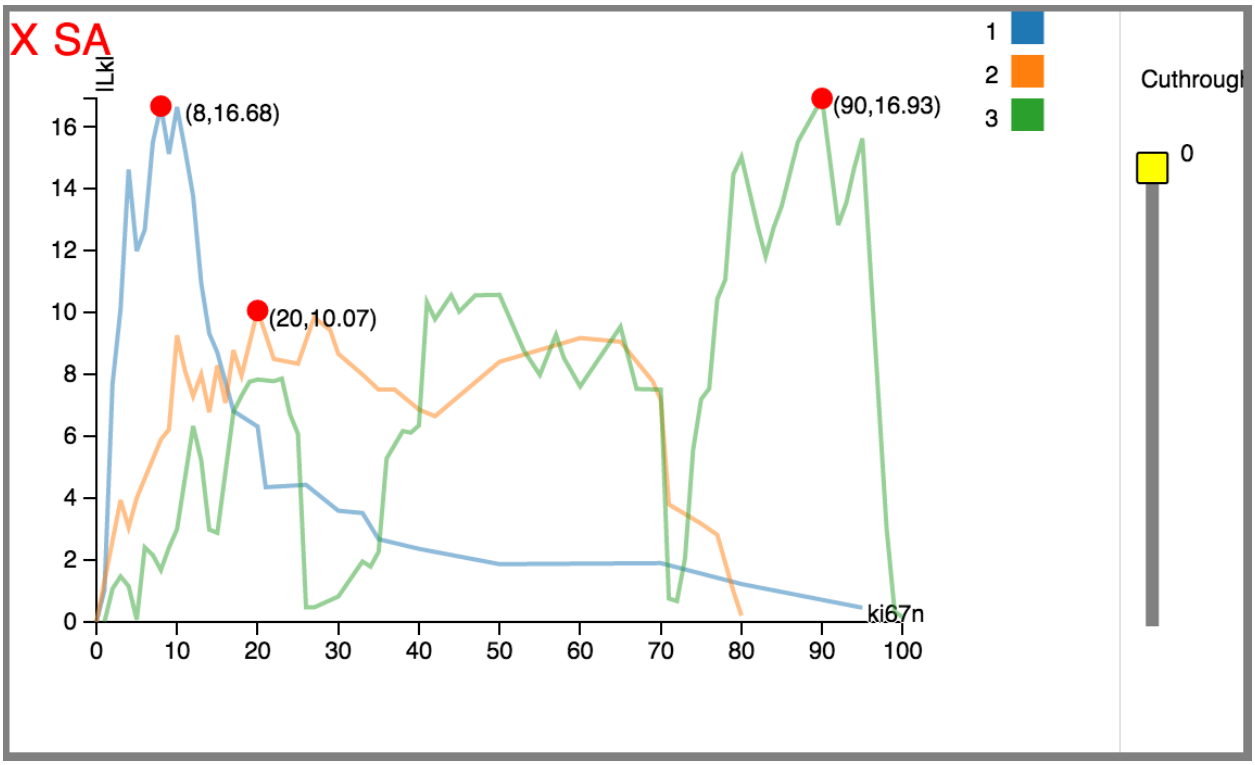


Figure (3.14) Visualizing optimal LRS cutpoint in different groups

context-dependent optimal cutpoints.

3.5.3 Realtime Visualization for Cut-through Analysis

Cut-through analysis is another common data mining practice in survival analysis. In cut-through analysis, a researcher assigns a cutpoint threshold r_δ for risk variable, and all the cases for those cases whose $R > r_\delta$ will be censored, no matter what their original censoring indicator values are. This operation might help to find a even better optimal cutpoint.

To conduct this analysis in pervasive software platforms, such as SAS, a researcher needs to modify the SAS macro and manually assign a threshold value, and then starts the time-consuming computation process. CutPointVis provides an alternative, visual approach. Figure 3.16 indicates the optimal cutpoint under the condition of cut-through analysis. Through GUI illustrated by figure3.16, a research can use the sliding bar on the right side to locate any cut-through point. While a researcher is operating a dragging action on the sliding bar, a realtime plotting of LRS using that point as censoring threshold will be presented.

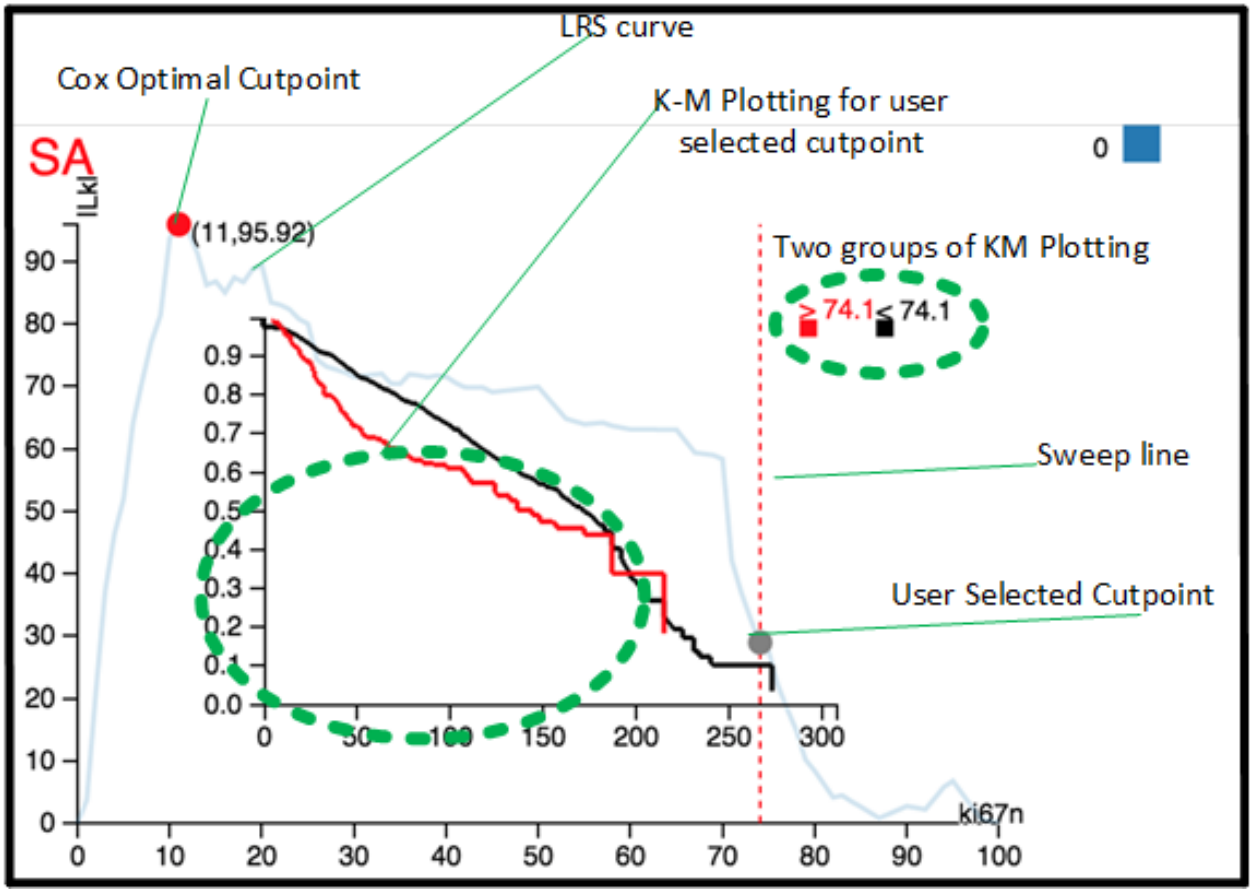


Figure (3.15) Realtime KM plot for optimal cutpoint determination

From this figure we can see that, when the cut through threshold is set to 57, we find another optimal cutpoint at $R = 20$, while the previous optimal cutpoint is at $R = 11$. Moreover, to compare different cut-through thresholds, CancerVis provides a "memory" function. If a researcher is interested in a cut through threshold but wants to compare it with other cut through thresholds, he can double-click on the current curve to "memorize" it. All "memorized" curves are presented in grey dotted lines, as indicated in figure 3.16.

3.6 CancerVis Usability Test: a Case Study

In this section we demonstrate the application of CancerVis with a comprehensive dataset composed of multiple studies stored within The International Cancer Genome Consortium (ICGC). This clinically dataset is composed of 288 pancreatic ductal adenocarcinoma

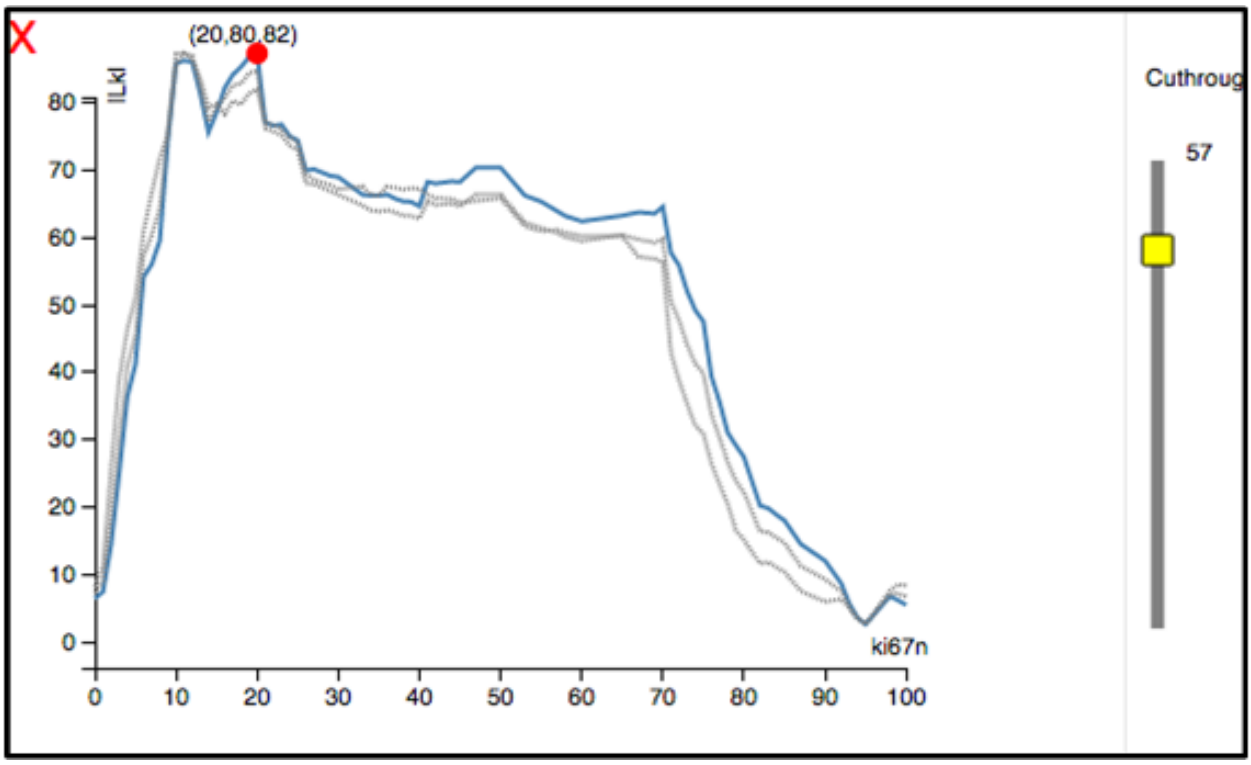


Figure (3.16) Visualizing optimal LRS cutoff point with different cut-throughs

cases. Each case has over 10,000 gene expression signatures and possess clinical and histological data such as grade and survival status. The median follow-up time of this dataset is 587.00 days. As these datasets come directly from research projects conducted with clinical patients, we consider this dataset to be an ideal one for biomarker investigation.

After examining the dataset with the CancerVis visualization tool and with statistical analysis. We have the survival summary based on "TumorGrade" as in table 3.1 and its survival rate is as in figure 3.17.

Table (3.1) Summary before stratification

Stratum	TumorGrade	Total	Failed	Censored	Percent Censored
1	1	42	6	36	85.71
2	2	157	90	67	42.68
3	3	89	61	28	31.46
Total		288	157	131	45.49

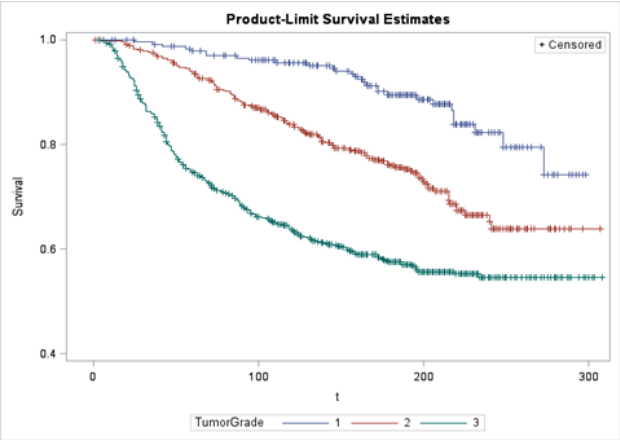


Figure (3.17) Original survival rate

We want to apply a new model to the groups divided by tumor grade. But before that we need to find optimal cutpoints in each grade itself. Therefore we use CutPointVis to visualize the cutpoints of the groups divided by tumor grade.

In this case, we use KI67 as the risk factor and follow-up-time as outcome. Censoring indicator is also recorded. Figure 3.18 is the visualization of 3 optimal cutoff points in three tumor grades.

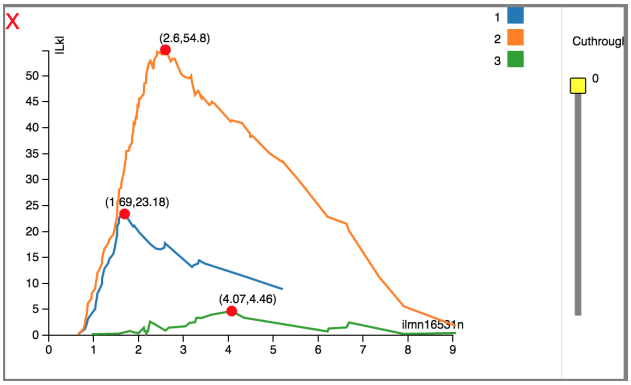


Figure (3.18) Visualizing optimal cut-off points in tumor grade groups

Within the dataset grade system we can quickly locate the optimal cutpoints: 1.69 (grade1), 2.59 (grade2), 4.07 (grade3).

If we only look at the ideal thresholds and the survival groups they create, we can create a model that fits the closest survival data into their own adjusted group. As indicated by figure 3.19.

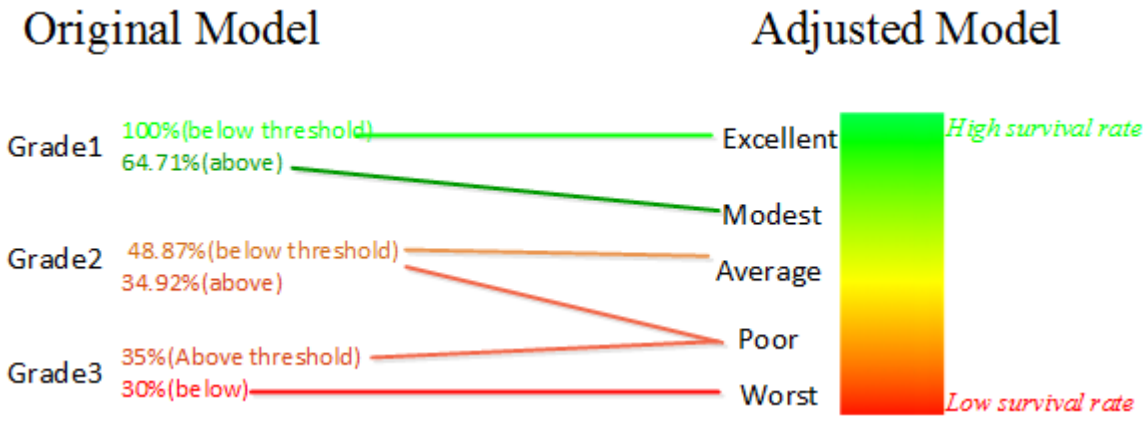


Figure (3.19) Remodeling three tumor groups

After adjusting the model, a better survival rate can be found in the newly stratified groups. The model fit is indicated by table 3.2. The survival summary is indicated by table 3.3. The adjusted survival rate is indicated by figure 3.20. The adjusted model exhibits a superior AIC to the original grading and allows for more accurate diagnosis.

This case study shows that CancerVis can improve biomarker research by providing an intuitive visual interface for finding optimal cutpoints. It can be used to supplement the traditional mathematical approach.

Table (3.2) Model fit after adjustment

	Adjustment	Original
AIC	1511.18	1523.3
SBC	1523.41	1529.41
-2LOGL	1503.18	1519.3

Table (3.3) Summary after adjustment

Stratum	Status	Total	Failed	Censored	Percent Censored
1	Excellent (NewGrade1)	25	0	25	100
2	Modest (NewGrade2)	17	6	11	64.71
3	Average (NewGrade3)	94	49	45	47.87
4	Poor (NewGrade4)	83	54	29	34.94
4	Worst (NewGrade5)	69	48	21	30.43
Total		288	157	131	45.49

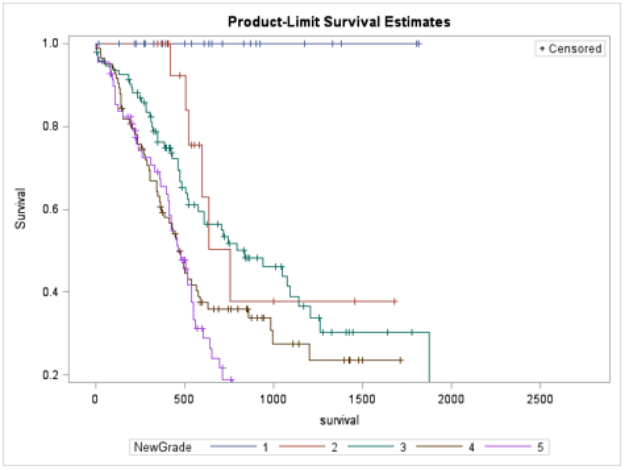


Figure (3.20) Survival of adjusted groups(plotted by SAS)

3.7 CutPointVis Verification and Case Study

3.7.1 Dataset and Workflow

In order to verify the usability of CutPointVis, in this section we conduct a case study to identify optimal hormone expression level thresholds which can most significantly stratify survival.

The datasets we use here are breast cancer mRNA datasets (GSE2034 and GSE7390) , which are available online at the Gene Expression Omnibus (GEO) [57] . We choose probe 205225_at (estrogen receptor, ER) and 208305_at (progesterone receptor, PgR) as risk factors. In this study we separately analyzed ER (probe: 205225_at) and PR (probe: 208305_at) expression

There are 286 cases in GSE2034 and 198 cases in GSE7390. The analysis workflow can

be descried as follows:

1.The researcher loads the dataset in cloud.

2. The researcher chooses the biomarker variable, outcome variable and optional censoring variables.

3. A LRS plot will be generated. The researcher explores the LRS, either inspect K-M plot or check a cutthrough plot(both in a realtime manner).

4. The researcher selects optimal cutpoint according to his own interpretation.

3.7.2 Exploration and Results

After loading GSE2034, we first choose an ER factor (205225_at in GSE2034) as risk factor and choose a survival factor(time to relapse or last follow-up), then we choose relapse as censoring factor. The LRS plot can be illustrated in figure 3.21. We can see that besides the Cox optimal cutpoint, there is less likely to be another context-dependent optimal points.

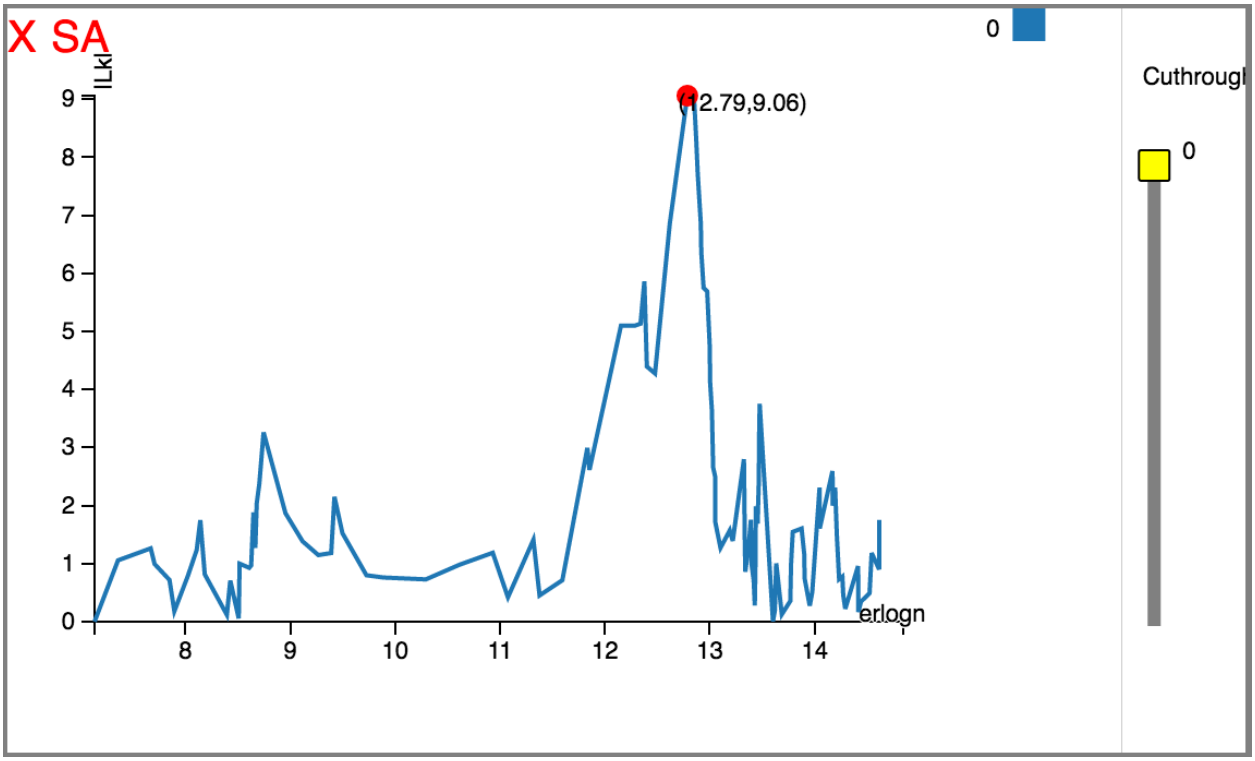


Figure (3.21) GSE2034 LRS plotting using ER as risk factor

In the next step we do cutthrough analysis. We slide the cutthrough threshold to 70 months (all those cases whose survival time over 70 will be censored) and we have an updated LRS plot, which is illustrated in figure 3.22. From this figure we can see that besides the theoretical optimal cutpoint, there is another potential cutpoint which worth investigation. Due to the different research contexts, different researchers may conclude on different cutpoints. CutPointVis provides conveniences for a researcher to switch and compare between different conditions.

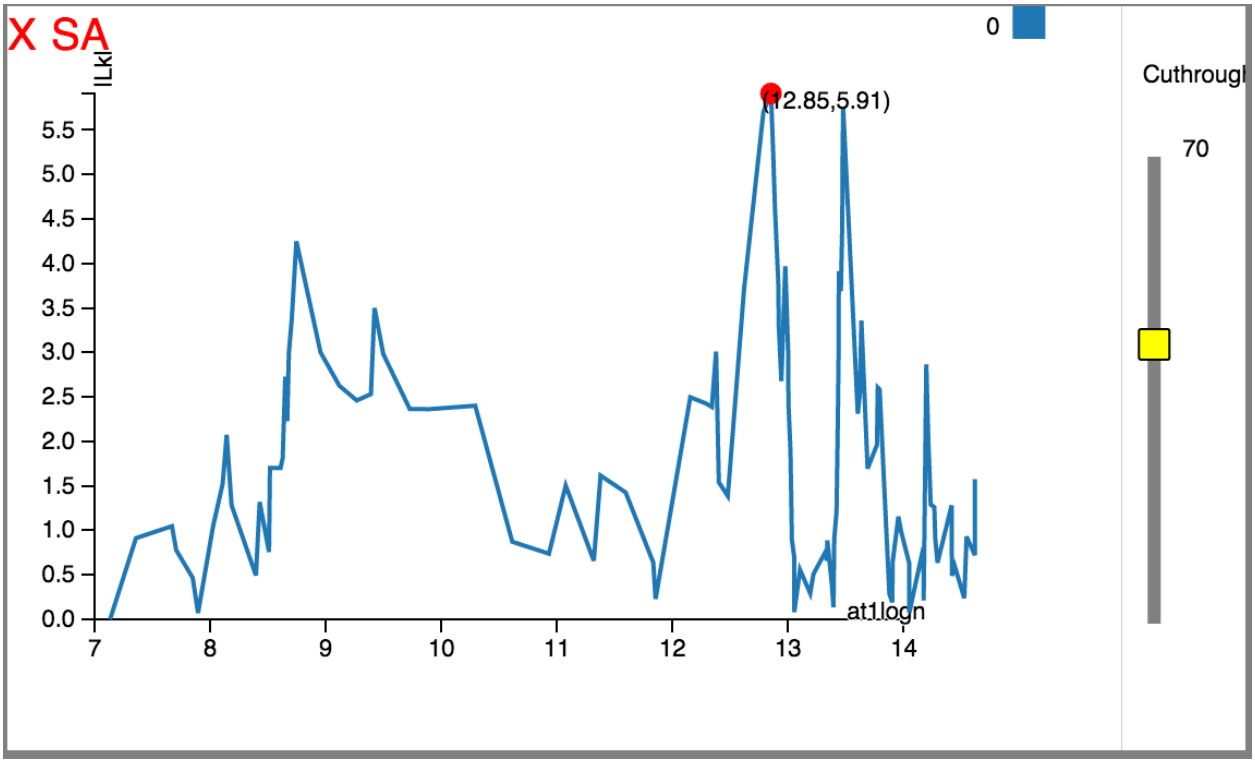


Figure (3.22) GSE2034 LRS plotting using ER as risk factor with cut-through

There are situations indicated in figure 3.11 where suboptimal cutpoint(s) that are also worth investigation. In the next step we user a PgR (20835_at) as risk factor and the LRS plot is illustrated in figure 3.23.

From this figure we can see that, although CutPointVis tagged the statistical optimal cutpoint with a red circle (where R = 6.08), but another suboptimal may be worth attention, which is the peak right beside the tagged red point. In this situation, a CutPointVis user

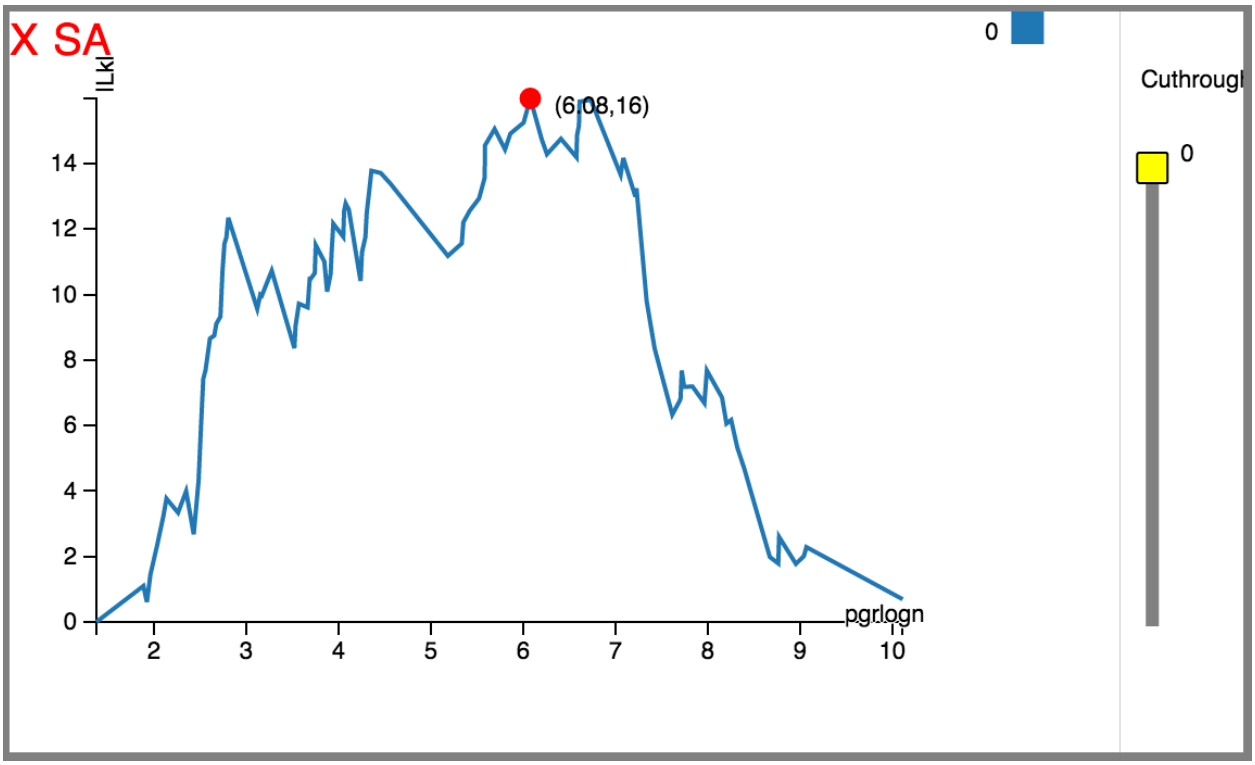


Figure (3.23) GSE2034 LRS plotting using PgR as risk factor

can easily operate the cursor (the vertical sweep line) to both two points, compare the K-M plot on-the-fly, then conclude on a context-dependent optimal cutpoint. The realtime K-M plot at these two points are illustrated in figure 3.24 and figure 3.25. It should be noted that, although the nuance of K-M plots between figure 3.24 and figure 3.25 is difficult to differentiate, the dynamic process of the KM plotting curve updates (when the cursor moves the cutpoint 6.08 to 6.79) can clearly tell a researcher the trend, such as if the gap is closing or growing. By this mode, CutPointVis provides the freedom for a researcher to conclude on his own version of optimal cutpoints.

As illustrated by this case study, CutPointVis would allow a researcher to:

- Identify if increasing hormones shows a monotonic survival trend, or if chosen thresholds are simply noise.
- Investigate all significant peaks to see if any are close to previously reported cutoffs.
- If ER/PR has bimodality at different expression levels [58]

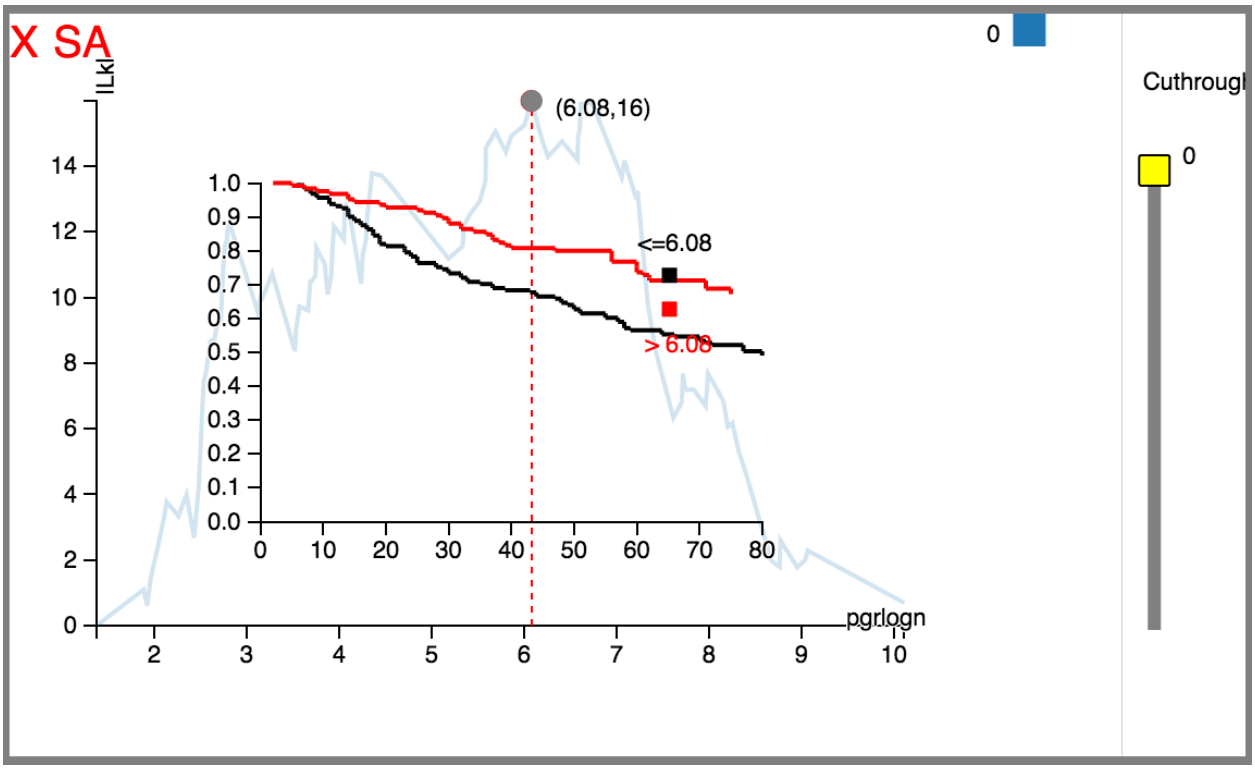


Figure (3.24) GSE2034 LRS plotting using PgR as risk factor(KM plot at R = 6.08)

- Potentially split cohorts into 3 survival groups (low, moderate, and high risk) based on hormone expression by identifying multiple extreme peaks

Due to the different research contexts, different researchers may conclude on different cutpoints. CutPointVis provides conveniences and the flexibility for a researcher to switch and compare between different conditions. For example, in figure 3.23 multiple PR thresholds exist with very similar peaks (6.08 and 6.79). If a researcher wanted inspect the survival stratification on these points with CutPointVis, he could simply operate the vertical sweep line to both points. The realtime KM plot at these two points are illustrated in figure 3.24 and figure 3.25.

For GSE7390, there is no PgR factor (20835_at), so we only choose ER factor (205225_at) as risk factor. The LRS plotting can be illustrated in figure 3.26. From this plot, we are interested in two cutpoints (green circle) besides the theoretical optimal cutpoint. By CutPointVis we quickly visualize the K-M plot at these three points, which are illustrated

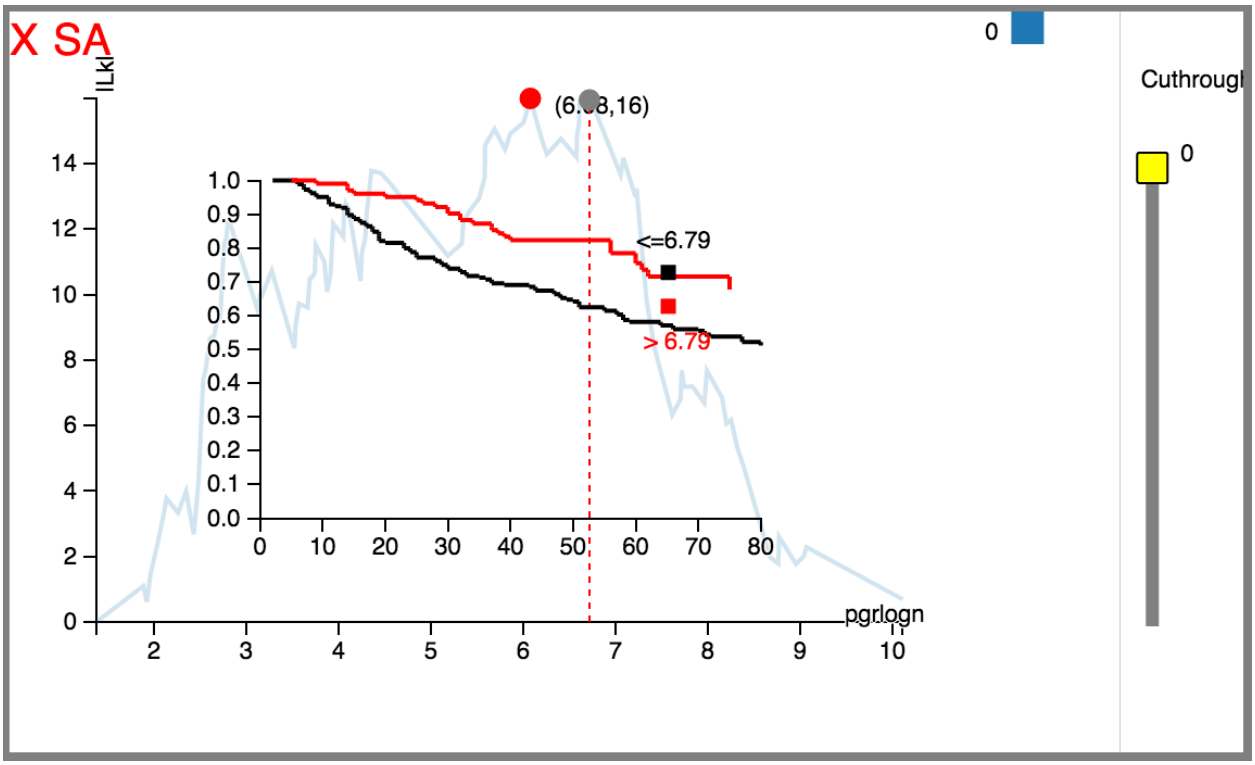


Figure (3.25) GSE2034 LRS plotting using PgR as risk factor(KM plot at R = 6.79)

by figure 3.27, 3.28, 3.29.

By the analysis of two public datasets, we demonstrate that CutPointVis provides a fast and convenient tool for cutpoint optimization. We believe CutPointVis improves the efficiency of biomarker analysis in cancer research and promotes the productivity in cutpoint optimization.

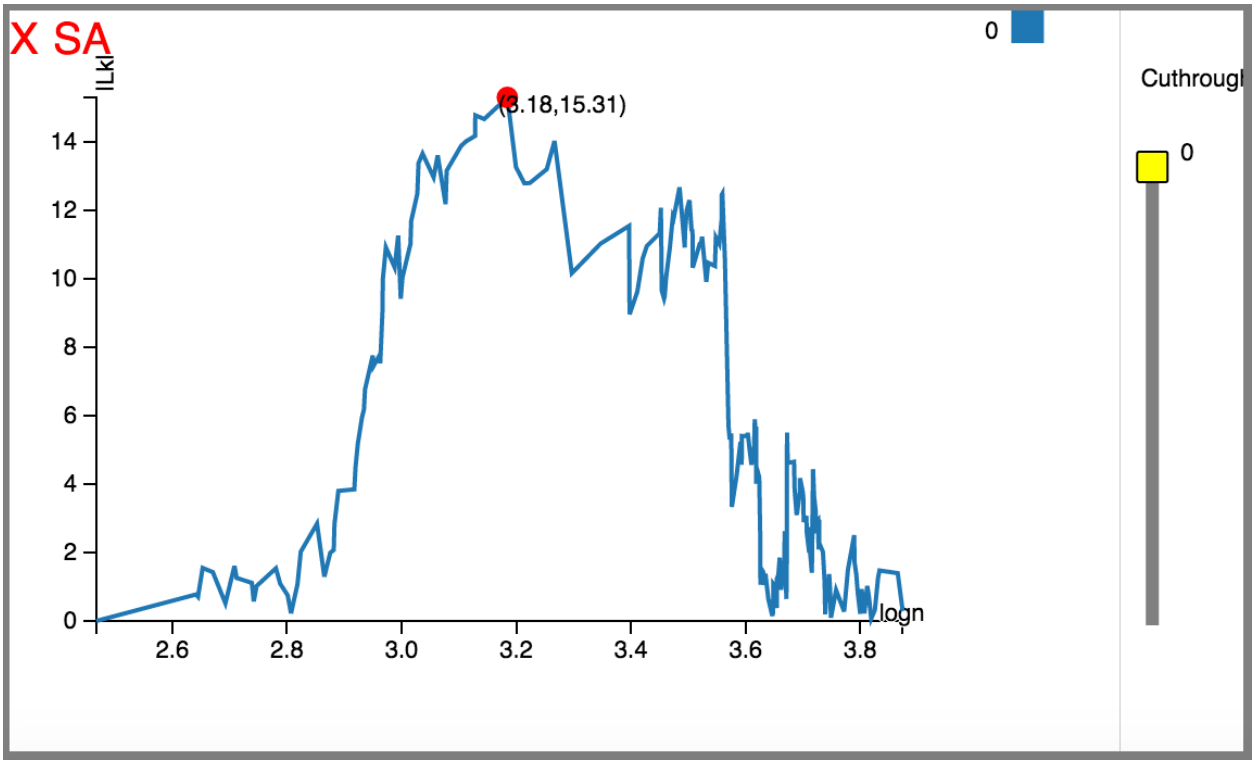


Figure (3.26) GSE7390 LRS plotting using ER(205225_at) as risk factor

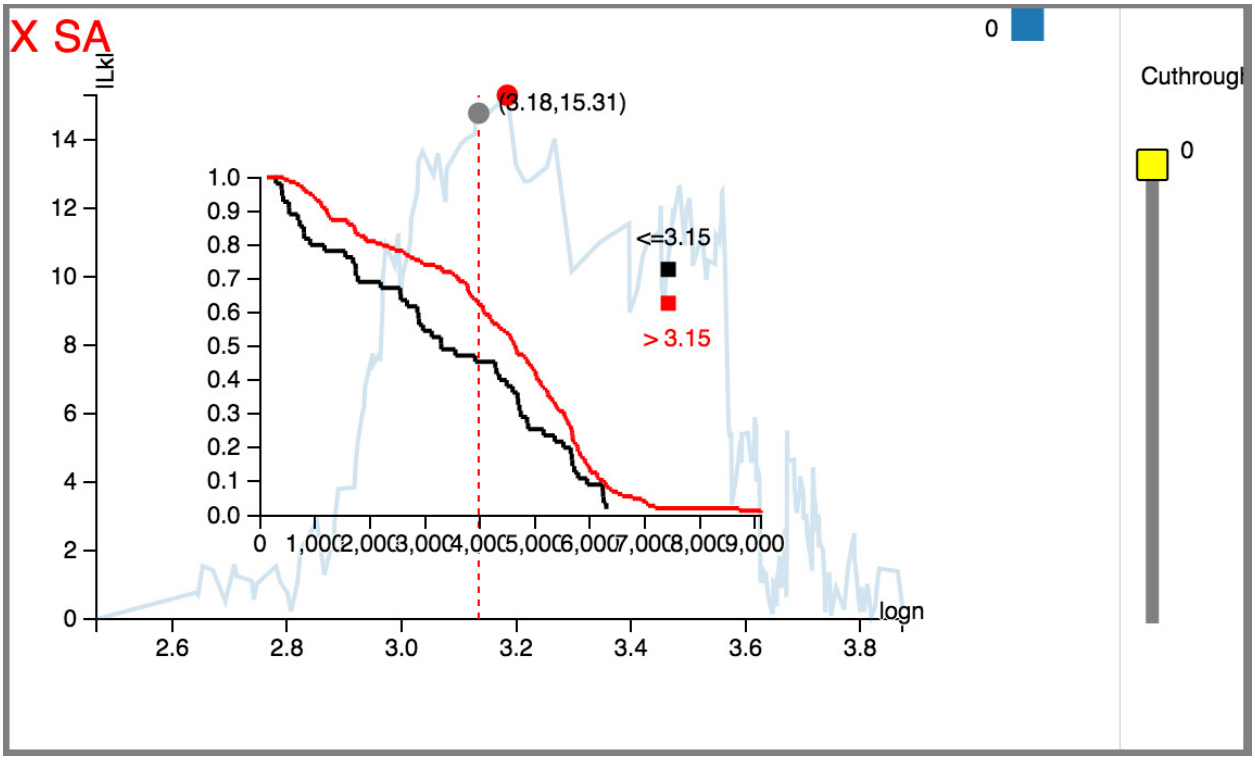


Figure (3.27) GSE7390 LRS and KM-plotting at R= 3.15

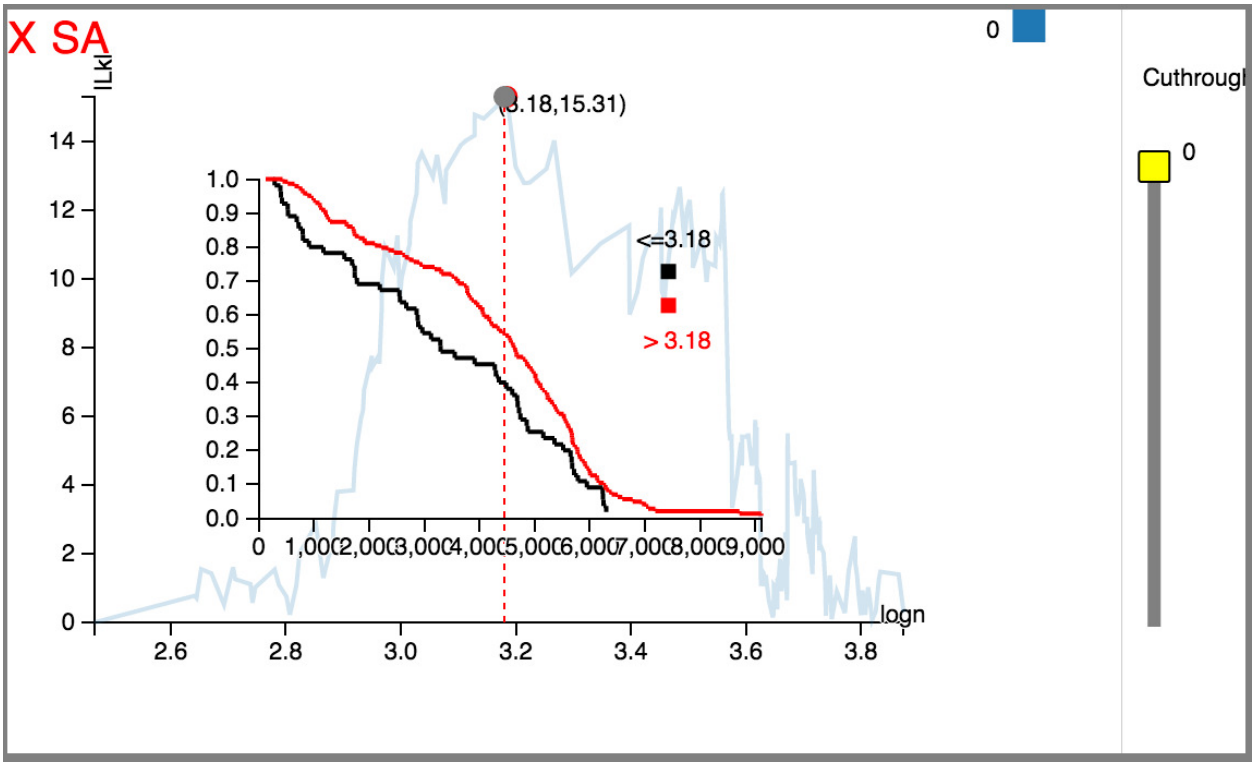


Figure (3.28) GSE7390 LRS and KM-plotting at R= 3.18

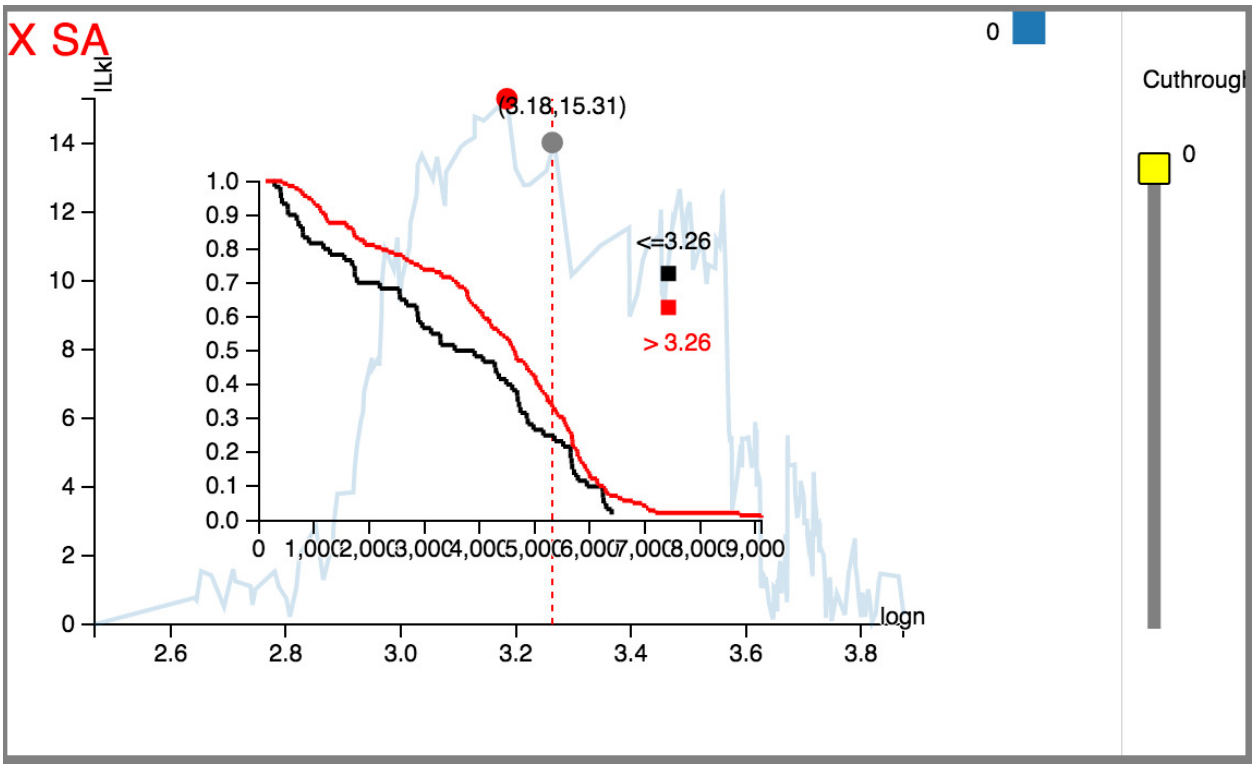


Figure (3.29) GSE7390 LRS and KM-plotting at R= 3.26

Chapter 4

TENNIS VISUALIZATION WITH WITH ON-DEMAND VIDEO REPLAY

4.1 Introduction

In the age of knowledge explosion, a sports consumer is exposed to a huge source of sports entertainment. With the help of Internet, retrieval of interested sports matches is no longer a problem. However, very often a person does not have enough spare time to absorb all the interested matches and keep updated. In order to get updates, a sports fan has mainly three options:

1. Full time match video.
2. Match report and statistics.
3. Match video highlight

It is often time-consuming to watch a relived full time video: a modern sports match often lasts hours. By reading a match report (with statistics), a fan is hard to get details of a match. A match video highlight is often produced by expert(s) in that sport. However, it lacks flexibility and is likely not satisfying fans' diversified preferences. A visualization platform, which can both present match statistics and offer match detail (video clips of tennis points) according to user interaction, can solve the aforementioned problem.

Based on our previous work [59], we developed TennisVis, a chart-based highlight extraction platform for tennis matches. With TennisVis, a user can get an overview (statistics) of a tennis match at first sight. At the same time, match details (video clips of each tennis point) are offered for user selective video play. Furthermore, TennisVis offers highlight recommendation, which presents a compact compile for top rated plays in a match.

The data input of TennisVis is twofold, a Shot-by-Shot textual description (S2STD) and a full match video. Match facts can be extracted and produced through text mining

the S2STD file. In order to identify a specific tennis point from a full time match video, the timing of each tennis point needs to be computed. Since timing information (time of each tennis point in the match video) is not available in S2STD, an Audio-based Tennis Rating Framework (ATRF) is developed to extract timing information of each single tennis point and evaluates a rating of each tennis play. The extracted video clips and ratings can be used for user on-demand video highlight play or automatic highlight recommendation.

Compared to other state-of-art work, our work distinguishes itself with following points:

1. We provide a straightforward platform for a user to visualize a tennis match, match facts are presented within a single graphical user interface.

2. Although one of our goals is extraction of video highlights, our solution does not involve any vision / image processing technique. This makes our solution computationally affordable.

3. Our solution offers an on-demand video play of match highlights, a user is able to choose to view any video highlight according to his individual interest.

4. TennisVis offers automatic highlight recommendation based on text mining and audio analysis.

Figure 4.1 illustrates the system architecture of TennisVis. Two types of inputs are S2STD and audio signal of full time match video. By text mining S2STD, basic match facts can be discovered and presented by main GUI. In Audio-based Tennis Rating Framework (ATRF), CCBHD algorithm detects and outputs the moments of tennis hits. After that, MSCA algorithm aligns tennis hits with tennis sets. Within a tennis set, MSCA algorithm aligns tennis hits with tennis games. Within a tennis game, MSCA algorithm aligns tennis hits with tennis points. By the end of ATRF, each tennis play is tagged with a temporal information (time of that play in full time video) and a rating information (spectator's reaction towards that play).

The remaining parts of this chapter are organized as follows: section 2 presents the related work. Section 3 presents a introduction to the TennisVis platform, which is the

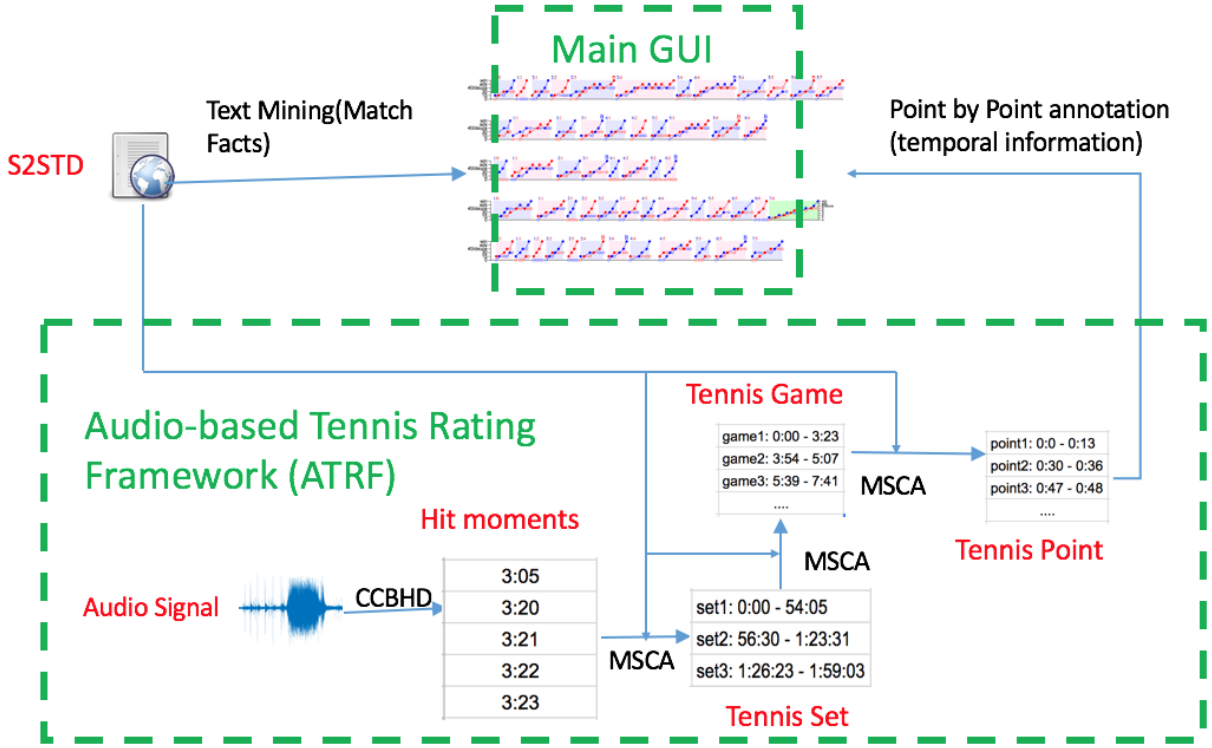


Figure (4.1) System Architecture of TennisVis

"Main GUI" part in figure 4.1. Section 4 presents CCBHD algorithm which detects tennis ball hits from audio signal. Section 5 presents MSCA algorithm which discovers set, game, play information for each detected ball hit. Section 6 presents a case study.

4.2 Related Work

4.2.1 Sports Data Visualization

Sports data visualizations research can be mainly divided into following categories: object detection and tracking [12, 60–64], semantic analysis (scene analysis and event detection) [15] and highlight summarization [17, 18]. Object detection and tracking are used mainly for sports performance measurement. Highlight extraction is popular in sports consumer since it offers a fast and compact version of replay video for those sports fans who do not have enough time to watch full-time relived video.

Semantics analysis in sports research refers to the extraction of video cues which can both reflect the structure of the video and be consistent with human understanding. Domain-specific knowledge are used in the semantic analysis. The extracted semantic events maybe trivial, since the semantic events are mainly defined in domain-specific knowledge, such as sports rules. For example, in a soccer game, these semantic events include fouls, corners, penalties, goal, substitution. As a sports fan, watching all these trivial video clips maybe as time-consuming as watching the full-length relived video.

[17] proposed highlight categorization that evaluates importance of a highlight and rank all the extracted highlights. This solution detects the slow-speed replayed part in the live video. [19] uses arousal level in replayed parts to rank highlights. Arousal level is calculated by audio energy and motion activity in corresponding events. [20] detects event in sports video through audio cues. It correlates audio signals with video frames, so that more accurate events can be detected.

4.2.2 Highlight Extraction

Highlight extraction / summarization [65] technologies has been widely researched to provide brief but significant parts of a sports match.

The state-of-art researches in this field are built on image/video analysis [66–68]. In [67], the proposed solution detects the repeated images in a sports video, since the repeated images are very likedly to be highligts of a match. In [68], the author developed solutions to compare edited videos with a full time video, so that highlights can be detected.

Among those research built on image/video analysis, many researchers introduce audio analysis to assist video analysis.

[20] processes the audio stream using audio strength level as indicators for candidate significant events. Each computed candidate is further verified by video frame processing. This research reduced the computation load of image processing by only examining the video frames in the time points output by audio processing. However, the algorithm in the first step (audio event detection) is not well elaborated. A peak detection algorithm is very likely

to have high probability of false alarms or false positives.

[19] uses video replay as the indicators of highlights. It extracts the replay clips from the video clips as highlight candidates. Then audio arousal signal and video frame analysis are used to screen candidates. At last, audio arousal level is used to rank the signification of each highlight. The computation cost of this research is expensive since it needs to image processing the who video stream. While replay parts of a video stream are highly probable highlight candidates in a video stream, there are other parts of a match which can also be part of highlights.

[69] combined audio analysis and video analysis to identify the significant parts of a match. It processes audio stream to output a candidate list of highlights. Then it employs text-detection technologies to verify categorized events. The audio signal analysis uses whistle and increased arousal in background noise as an indication of possible highlights. The text-detection technology interprets specific texts from TV screen and annotate audio outputs. The overall highlight quality depends on the quality of the whistle/arousal increasement recognition, while the TV broadcast pattern limitation (text display after each match event) makes it not robust to many sport highlight extraction scenarios.

[70] researched the highlight extraction in soccer matches. It applies domain knowledges in image motion detection to reduce computation resource usage. Meanwhile, replay detection is also used an indicator of significance of a highlight candidate.

Instead of using audio as highlight cues, [71] used text webcast contents as source information. Since webcast texts contain both event literal description and timestamp of that event, it is easy to align each text live event with its corresponding video part. The calibration of event time is still accomplished by video analysis.

It should be noted that all the aforementioned research and many other researches [72–74] employed image/video processing technique to analyze frames, audio information is used as an auxiliary dimension of information. In TennisVis, we do not adopt any image/video processing technique. We consider audio stream and text commentary are good and complementary sources of match information.

4.3 TennisVis Visualization Platform

4.3.1 Overview of TennisVis

In TennisVis, the data input are video stream of a match and its text details of that match. For the text data source, we selected TennisAbstract [75], an open crowdsourced project aiming collecting shot-by-shot data (S2STD) for professional tennis matches. As of May 2017, there are 3,128 matches collected. We selected full match video with no commentator as our video input. It is not hard to find required videos from Youtube.

The following two paragraph illustrates two typical entries in S2STD collected from 2011 French Open.

```
Roger Federer | 0-0 | 0-0 |15-0 | 1st serve wide, fault (wide).
2nd serve wide; forehand return crosscourt (deep); forehand down the
middle; backhand inside-out; backhand down the middle; backhand inside-
out; backhand down the middle; forehand crosscourt; forehand down the
middle; forehand inside-out; forehand crosscourt; backhand crosscourt;
forehand crosscourt; backhand down the middle, forced error.  (13-
shot rally)
```

```
Rafael Nadal | 2-0 | 2-2 | 40-30 | 1st serve wide; backhand re-
turn down the line (very deep); backhand down the middle; forehand
approach shot inside-out; forehand crosscourt; backhand volley down
the line; backhand slice down the line, forced error.  (6-shot rally)
```

We use vertical bar "|" to represent big space between fields in real documents. It can be seen from these two entries that, the description text is well structured and can easily be text mined. We call the first four fields as the basic match facts (e.g. "Roger Federer |0-0 |0-0 |15-0"), since these fields clearly state server, set score, game score, and point score. The fifth (also last) field is a manual text description of that point. This field offers a detail description of each shot (e.g. forehand or backhand, middle or crosscourt). Therefore, by

carefully mining the last field, TennisVis gets a detailed understanding of a tennis match.

Figure 4.2 illustrates the main GUI of TennisVis. The main GUI consists of two parts. Part A presents all the basic match facts, which all the data is parsed from those basic fields in S2STD. Part B offers flexibility and diversity for a tennis fan. Since different fans may have different preferences, TennisVis offers a query/filter section in part B of main GUI. A user can select all the points in his watch preference to view. All those points will be highlighted in Part A. Besides the visualization of match fact charts, TennisVis offers an on-demand video play function. With a user-click, each point on the chart can be fast forwarded in full-time video and played.

In the remaining parts of this section, we present the details of two parts in main GUI and the function of on-demand highlight play.

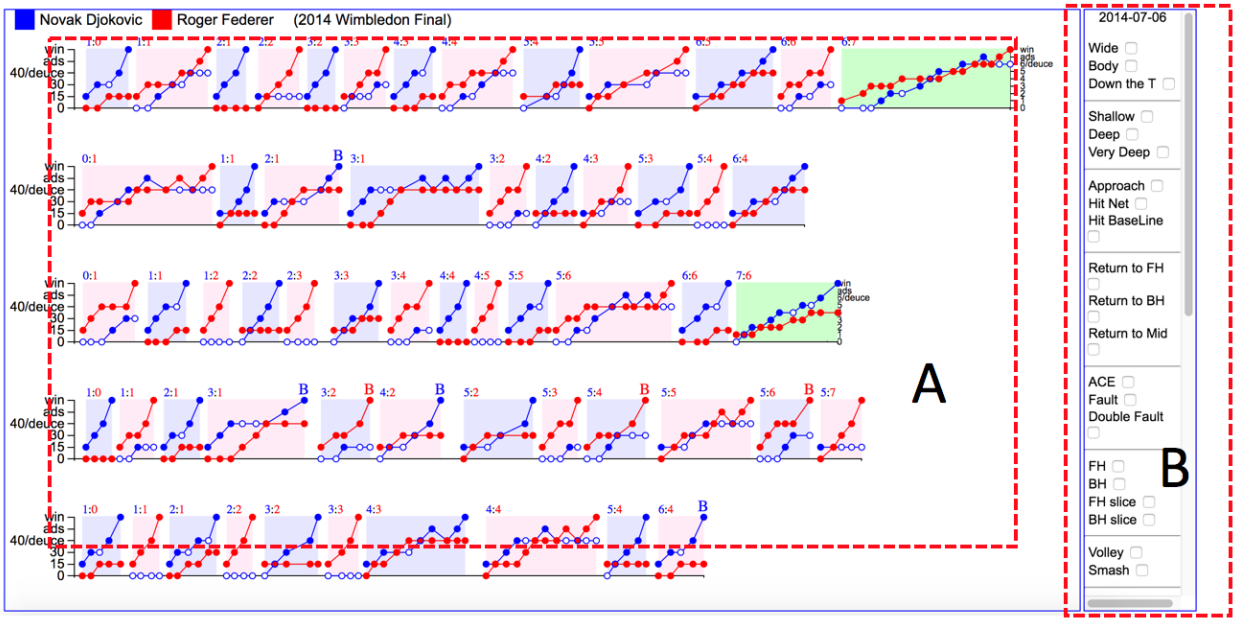


Figure (4.2) TennisVis GUI

4.3.2 Match Facts Charts

After parsing a S2STD file, a group of charts will be rendered in part A of TennisVis main GUI. The match fact part presents each match set with a axis-ed chart. As illustrated

by figure 4.3. Performance of two players score are plotted as two curves. Each game is represented as one group in axis-ed chart. Game score is indicated on top of each group.

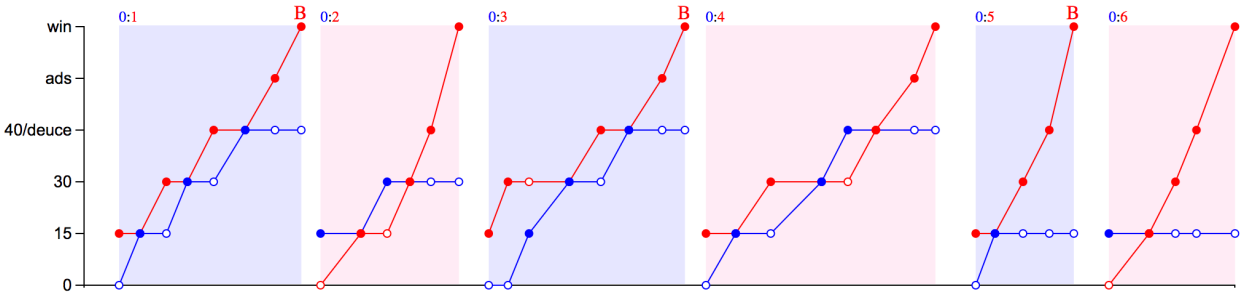


Figure (4.3) A tennis set visualization

A single game chart is illustrated in figure 4.4. In each game chart, the background color indicates the player who served in that game. For example, in this chart, the curve of blue represents the score of the player who served in this game. At each tennis point, two points is plotted, which represent the point score for both players. For example, after third point in this game, the score is "30:15", with score of server to be "30". Meanwhile, at each point, an empty circle indicates that player lost that point.

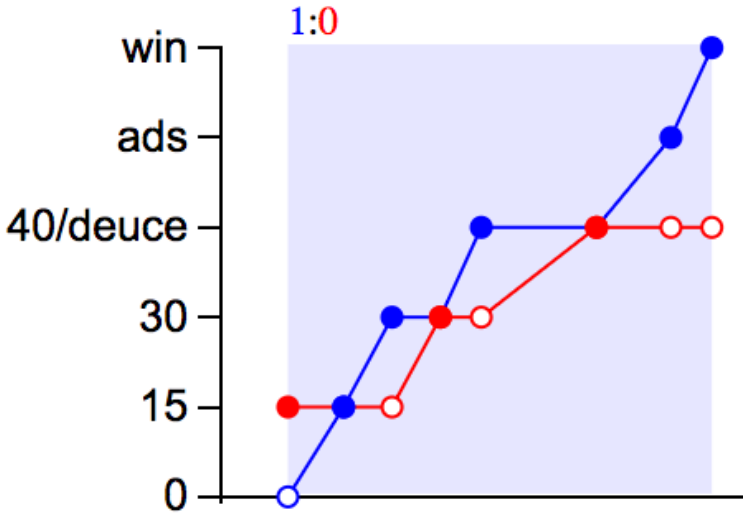


Figure (4.4) Visualization of one game

Although this visualization is simple, an experienced fan can still find clues of the rating

of a tennis point. For example, figure 4.5 illustrates the charts of two games. There are three deuces in the left game while the game on the right is a love game(40 - 0). It is very likely that the left game is more worthwhile watching than the right game.

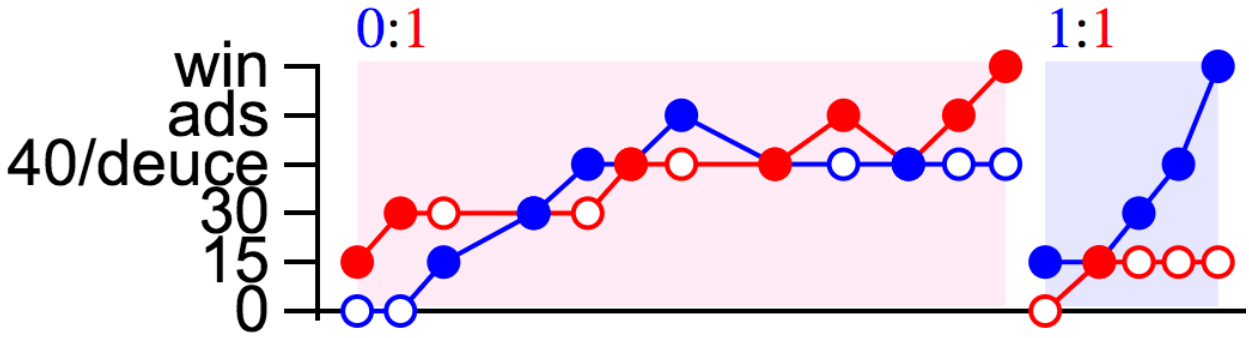


Figure (4.5) Visualization of two games

4.3.3 Match Detail Query

Tennis fans get little useful information from the basic match facts - they can get this information from any match report within one minute. What excites them more is the match details, such as Volley, Ace, many-shot rally or drop-shot. Unfortunately, besides watching full time video, it is hard for tennis fans to watch all the preference details.

TennisVis offers a query module (Part B in figure 4.2) for its users to query all the details of a match. By text mining the S2STD file, TennisVis summarizes each tennis point with following attributes in table 4.1.

A tennis fan can query any combination of aforementioned attributes, TennisVis returns all the matched points with highlighted indication on the basic match fact presentation. Figure 4.6 illustrates a query and results. In this figure, a user queried all points which are *drop* shots or *forced errors* or *number of rally shots more than 10*. The corresponding tennis points are indicated with black dots in charts on the left in real time. Meanwhile, a statistics of the query of each set is displayed on bottom of each set chart.

Table (4.1) Attributes of tennis points

Wide	Body	Down the T
Shallow	Deep	Very Deep
Approach	Hit Net	Hit BaseLine
Return to FH	Return to BH	Return to Mid
ACE	Fault	Double Fault
FH	BH	FH slice
BH slice	Volley	Smash
Drop shot	Lob	Half-Volley
Swing-Volley	Winner	Unforced Err
Forced Err	number of rally shots	Break Points
Game Points	Set Points	Match Points

4.3.4 On-demand Video Clip Play (ODVCP)

After a user finds out all his interested parts in a tennis match, the next step is to watch these parts in the match video. However, in a traditional way, given a full time match video, a user has to rewind forward and backward many times to locate to a specific time point which corresponds to a specific tennis point. This process is both time-consuming and troublesome. It often makes a fan turns into watching full time video or some officially released highlight video compiles. In TennisVis, we developed a function, namely on-demand video clip play (ODVCP), which offers on-demand video play of each single tennis point. If a user is interested in any tennis point, a simple click on that point will play its corresponding video part in full time watch, as illustrated by figure 4.7. This saves a user much time and offers the flexibility of self-selection, customized highlight watching.

ODVCP distinguishes our work from other related work. In the next section, we present our solution to ODVCP.

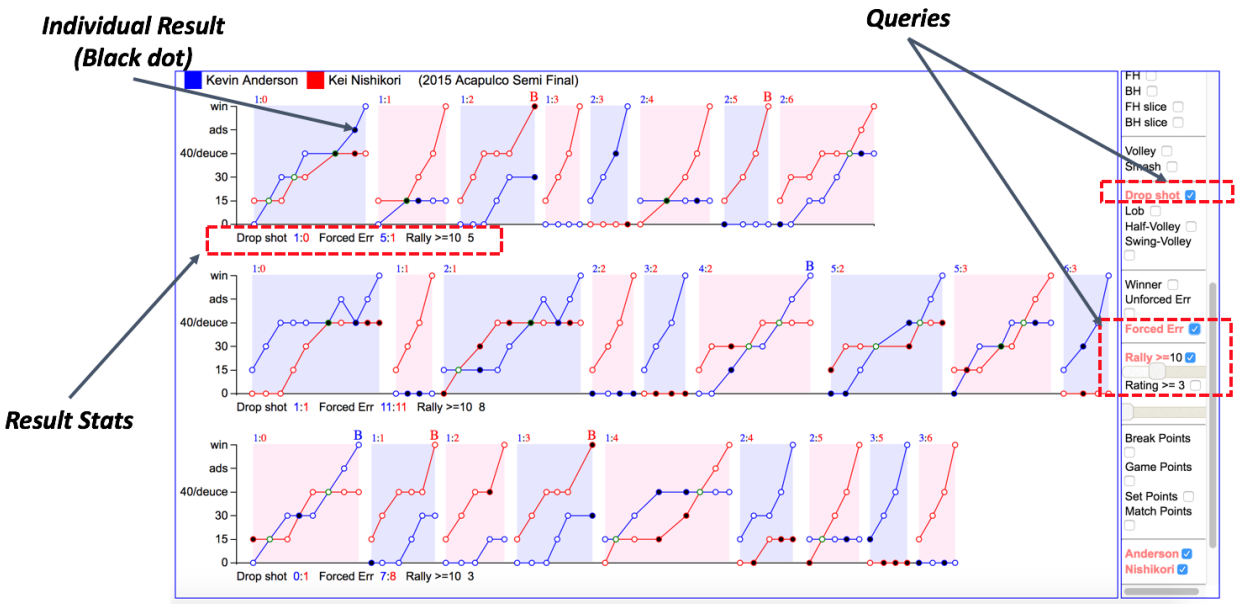


Figure (4.6) Query and results

4.3.5 Match Highlight Recommendation

Besides the statistical side of match facts (e.g. a volley shot, N-shot rally), TennisVis offers a rating for each individual play. Since ATRF has computed the timing information of each play, TennisVis uses spectator's applause length following that play as an indicator for that play.

Figure 4.8 illustrates TennisVis with rating indicator. In this figure, circle for each point is given different sizes. Larger size indicates higher ratings.

4.4 Ball Hit Detection from Audio Signal

In this section we present tennis ball hits detection mechanism in ATRF framework. We present the theory and design of Cross-Correlation Based Hit Detection (CCBHD) Algorithm.

In order to realize the on-demand selection, temporal information is supposed to be fed into the GUI so that the corresponding video part can be played. Unlike other state-of-art research, we choose match audio streams as the source from which temporal information is

Figure (4.7) On-demand highlight selection

retrieved. We name this process as a synchronization task, which annotates each text point with both timing and rating information.

In a tennis match, ball hits can be considered as meta information, many other higher level semantic events can be inferred from ball hits [76]. Figure 4.9 illustrates a typical ball hit in audio signal. It should be pointed out that, the signal in figure 4.9 is sampled in an ideal environment, where the noise is low and can easily be reduced by a finite impulse response (FIR) filter. However, in a tennis match, the spectators may introduce significant noise to audio signal. Figure 4.10 illustrates a piece of audio signal in a tennis grand slam match. In this figure, spikes are ball hit events while the red dotted rectangle illustrates the background noise from spectators. A closer look of a single ball hit is illustrated in figure 4.11. In this figure, red dotted rectangle 1 tags panting of a player, red dotted rectangle 2 tags a ball hitting event, red dotted rectangle 3 tags an event that a ball hits net. Other baseline parts can be considered as background noise in the stadium (spectators are disciplined to keep quite during the first hit and last hit of each point). A naive method is difficult to

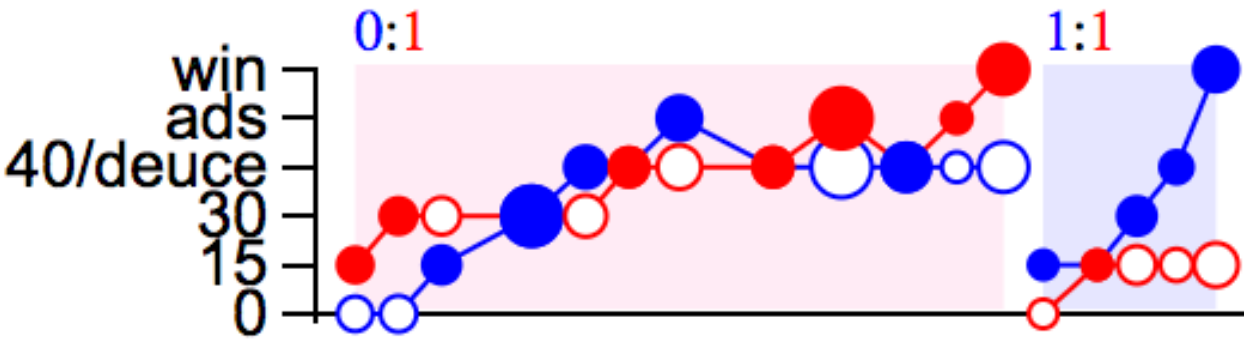


Figure (4.8) Recommendation of tennis points

distinguish the ball hit from other unrelated events.

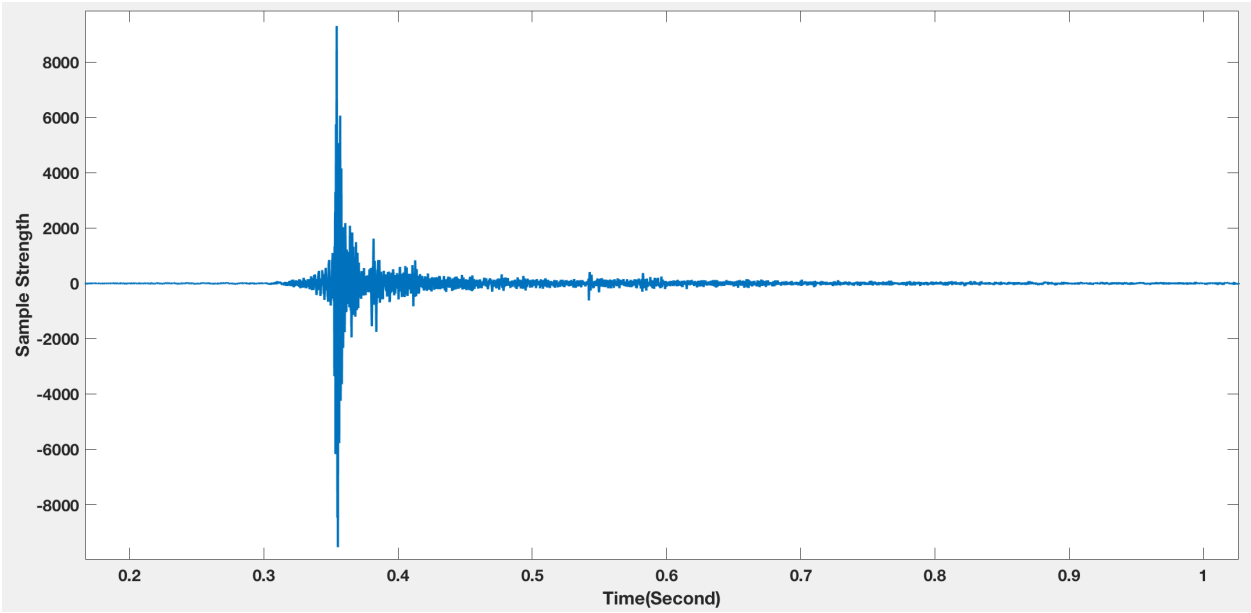


Figure (4.9) Ideal tennis ball hit

After studying various real time data, we find that most ball hit waveforms are similar to each other. Based on this observation, we take the assumption that a referential waveform is similar to most of ball hit waveforms. Then the ball hit detection can be transformed into computing the similarity between a real time signal and a referential signal. Suppose that the real time signal in figure 4.11 is denoted by $m(t)$, t is time variable. As stated before,

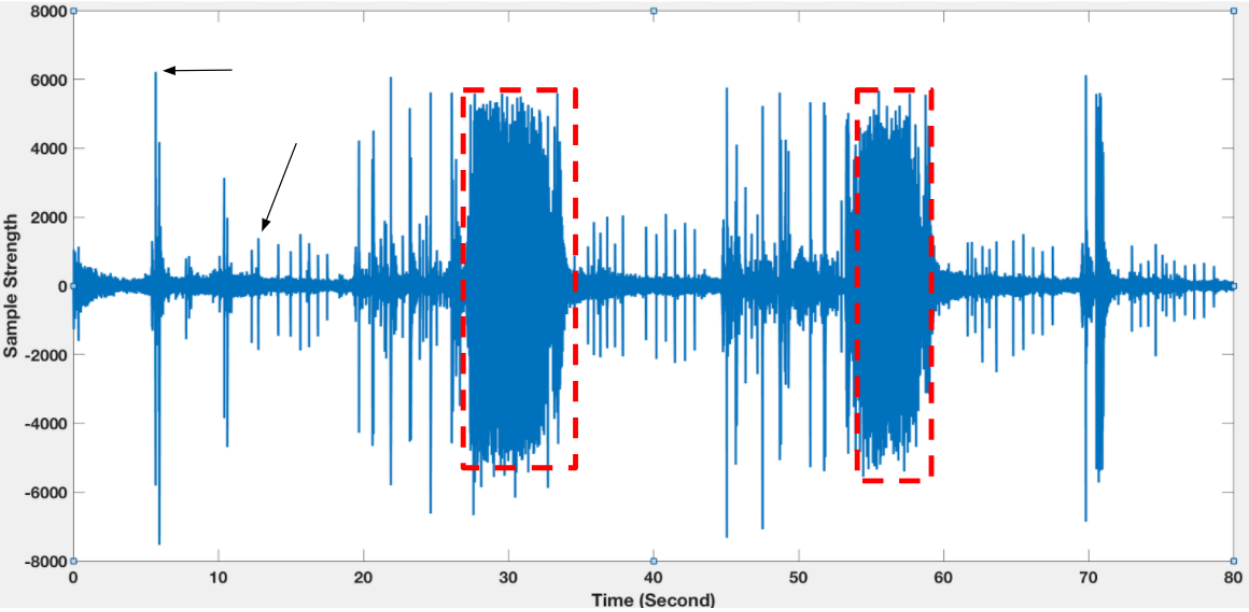


Figure (4.10) Audio signal of a match piece (80 seconds)

$m(t)$ is composed of two parts, ideal signal and background noise. As a result, $m(t)$ can be expressed by equation 4.1:

$$m(t) = s(t) + n(t) \tag{4.1}$$

In equation 4.1, $s(t)$ is an ideal ball hit signal, $n(t)$ is background noise. We adopt the assumption that $n(t)$ is a white noise. It should be noted that, although in figure 4.10 the spectator noise level (pieces in two red dotted rectangles) is much stronger than the white noise in the rest part, it does not violate our assumption in equation 4.1. The reason is that we are only interested in event detection in the play time (when two players are hitting balls against each other), the spectator noise can be easily ruled out with a moving average threshold.

Evidently, $s(t)$ is zero when there is no ball hitting event in the signal. Denote $h(t)$ to be the referential ball hit waveform, $'*'$ operation computes the similarity between two waveforms. Then the similarity between a ball hit signal ($m(t)$) and a referential signal ($h(t)$)

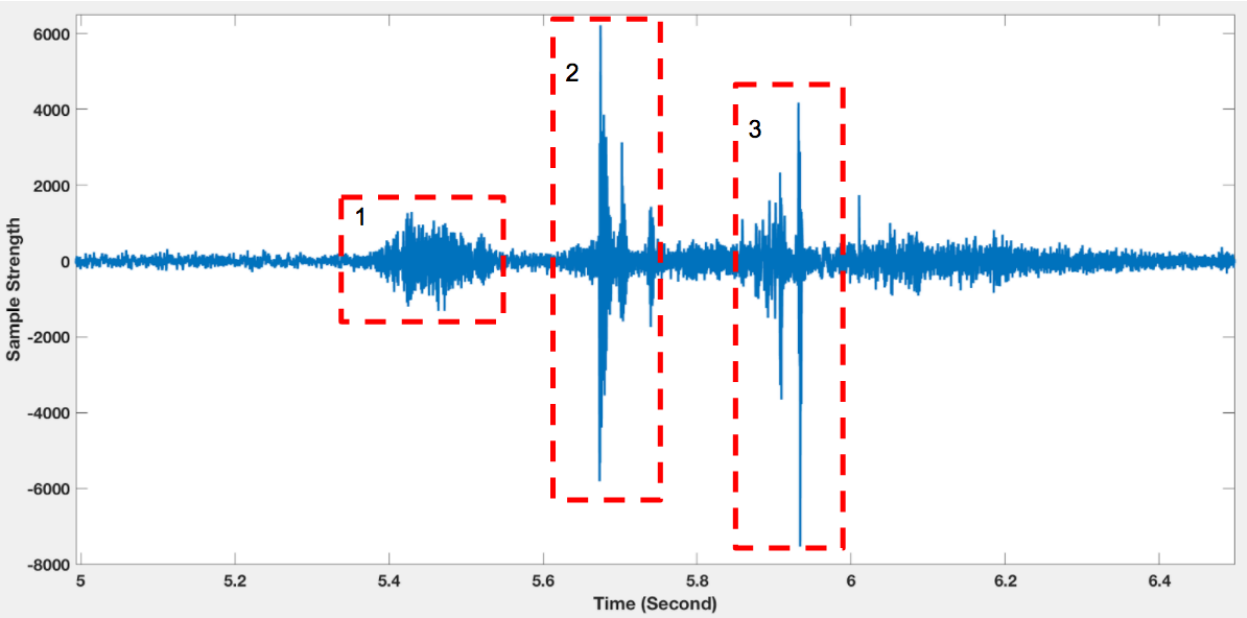


Figure (4.11) Audio signal of a ball hit

can be expressed as follows:

$$m(t) * h(t) = (s(t) + n(t)) * h(t) = s(t) * h(t) + n(t) * h(t) \tag{4.2}$$

In equation 4.2, as we assume that $n(t)$ is white noise, the $n(t) * h(t)$ will output a random variable with average at zero, because white noise is not similar to any predefined referential signal. Therefore, the output is mainly determined by the component $s(t) * h(t)$. In equation 4.2, we adopt cross-correlation [77] coefficient ρ to define the $'*'$ operation, as illustrated by equation 4.3.

$$\rho_{fg} = \frac{\sum_{t=-\infty}^{+\infty} f(t)g(t+n)}{\sqrt{(\sum_{t=-\infty}^{+\infty} f^2(t))(\sum_{t=-\infty}^{+\infty} g^2(t+n))}} \tag{4.3}$$

In equation 4.3, g is a predefined referential waveform which lasts N sample duration, f is a discrete signal of time variable t. In equation 4.3, the inner product of f and g is computed at each sample point. If there is a similarity between g and a piece of f, the value

of $(\rho(f * g))$ will be maximized, which indicates a high probability a ball hit event.

A key step for applying 4.2 to ball hit detection is the choice of predefined referential signal $h(t)$, which should be similar to most of tennis ball hit waveform. Here we use a heuristic method to derive a referential waveforms. We select five typical ball hit waveforms, then we apply a moving average filter to each original signal. One of the referential signals is illustrated in figure 4.12.

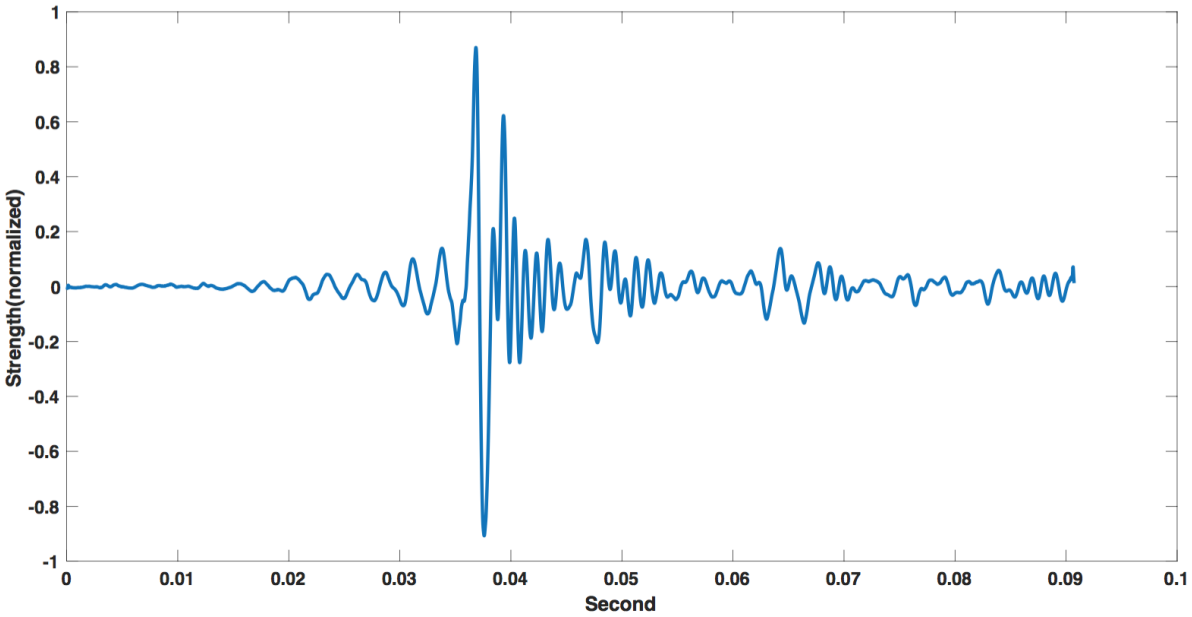


Figure (4.12) Selected referential signal

After the referential signal is determined, we use a sliding window to apply the cross correlation operation to the real time data, the results can be illustrated by figure 4.13 and figure 4.14. From figure 4.13, we can see that almost all the ball hit spikes are correlated with a cross-correlation spike, while the cross-correlation spike is smooth and easy to detect. Furthermore, at the two red dotted rectangles, which are noise caused by spectators, the cross-correlation indicates a very low coefficient, it is easy to rule out spectator's noise based on cross correlation results. From figure 4.14, we can see that, the ball hit corresponds to a higher cross-correlation result, while the panting and net scratching part can be ruled out by choosing a proper threshold in the cross correlation results.

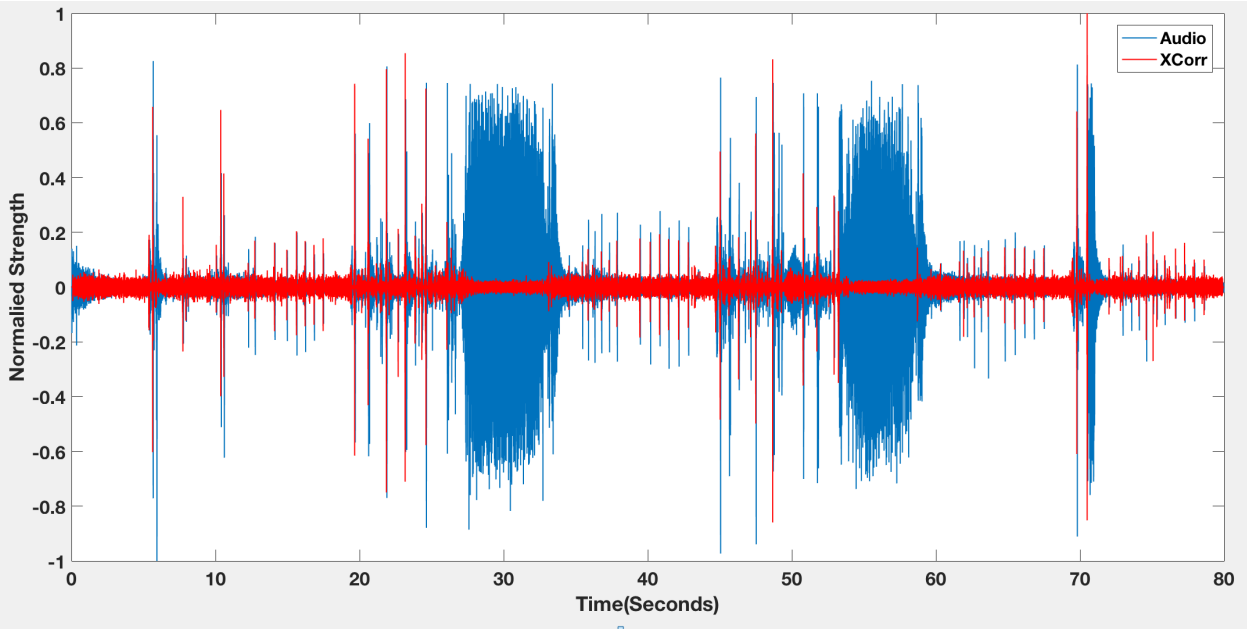


Figure (4.13) Cross-Correlation results of a piece of match signal (80 seconds)

In this section we presented our solution to detect tennis ball hit by applying a cross-correlation operation on match audio signal. It should be noted that, the accuracy of ball hit detection is around 70% (Illustrated in case study section) with missing detection (false negative).

4.5 Identifying Tennis Plays : MultiSet Counting Algorithm

As illustrated in figure 4.1, after hit moments are computed, TennisVis seeks to synchronize hit moments with textual sets, games and plays. In this section, we present our solution, MultiSet Counting Algorithm(MSCA) , to the synchronization task, which allocates audio hits into tennis sets, games and points. We developed a mathematical model for the synchronization task and we introduce our method to make the allocation algorithm computational affordable.

For the convenience of presentation we introduce/define some concepts/terms in our context. We define the algorithm to the synchronization task as τ. According to [78], a tennis match is composed of **POINT**s, **GAME**s and **SET**s. To avoid confusion, we refer

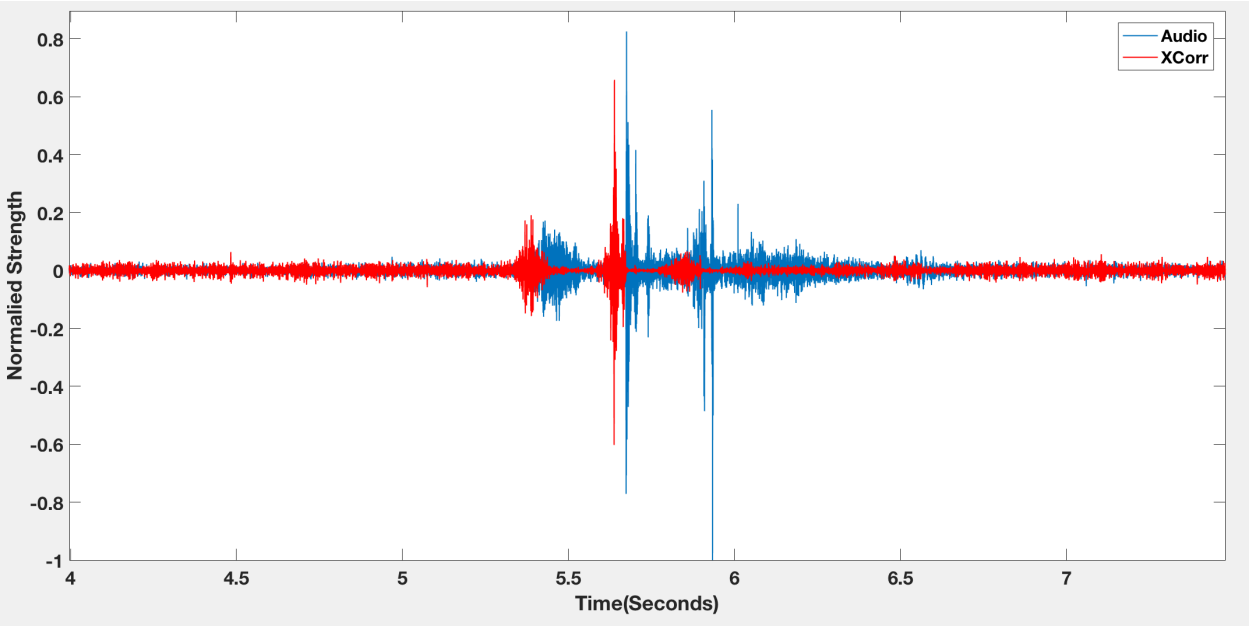


Figure (4.14) Cross-Correlation results of a ball hit

each item as tennis game, tennis set, and tennis point. A tennis match is composed of tennis sets. A tennis set is composed of tennis games. A tennis game is composed of tennis points. Besides the concept of tennis set, tennis game and tennis point, we introduce one more concept, tennis **PLAY**. A tennis point is composed of one or more tennis plays. Each tennis play starts with a service. The number of plays in each point may be more than one because of first serve failure or Let (A let occurs when a legally delivered ball lands in the cross-court service box having touched the net cord).

4.5.1 Problem Statement

From the audio signal, ball hit events can be detected, meanwhile, time of each audio ball hit can also be produced. Due to the inaccuracy of detection, there is not a one-one mapping between audio hits and text points. A demonstrative example is given in figure 4.15. In this figure, a tennis game data is presented. The left part is a time axis on which all the detected events are tagged. It should be noted that the position of each hit is tagged proportionally to time elapse. The light solid blue dots are correct detection (based on manual checking ground truth video.), the red empty circles indicate missing detections

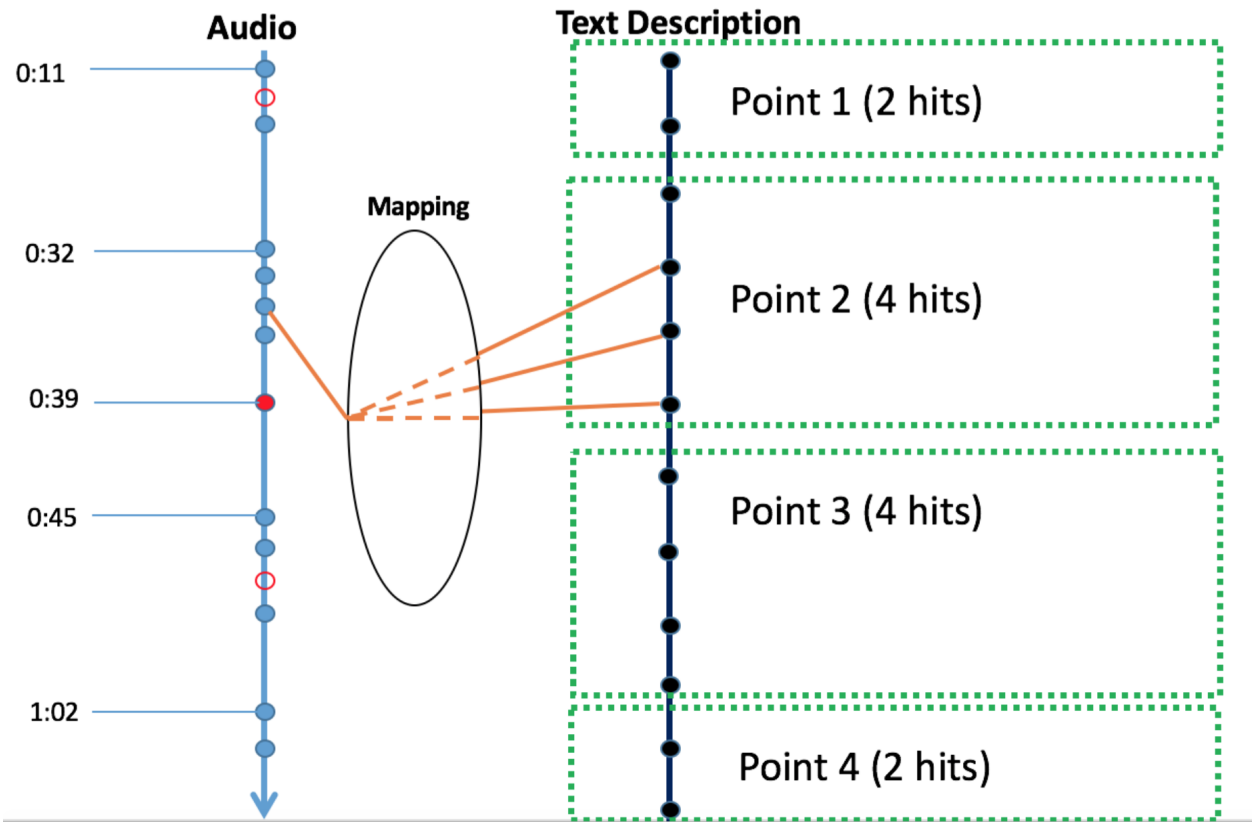


Figure (4.15) Problem statement

(false positive). It should be noted that we ignored false detection here, the reason is that the probability of false detection is low and as stated in previous section, we can rule out almost all false detections by checking background noise level. The right part are the match facts parsed from a S2STD match description. By text mining S2STD contents, number of hits of each point can be determined.

The synchronization task is now how to put hits in the left into the each tennis play / bag on the right side. Suppose there are n hits (on the left side of figure 4.15) represented in the following sequence:

$$H = \{a_1, a_2, ..., a_n\}$$

, while there are m plays represented in the following sequence:

$$B = \{b_1, b_2, ..., b_m\}$$

on the right side (Play i has b_i hits). For the case illustrated in figure 4.15,

$$\begin{aligned} &a_1 = a_2 = ...a_{12} = 1 \\ &b_1 = 1, b_2 = 2, b_3 = 3, b_4 = 4, b_5 = 2 \end{aligned} \tag{4.4}$$

Equation 4.4 holds because there are 12 detected hit events and the number of hits in each play is (1,2,3,4,2).

Our solution is supposed to allocate each hit (a_i) into one of m bags. We model a possible allocation scheme (combination) C as follows:

$$C = \{c_1, c_2, .., c_m\}$$

where each element c_i is a subset of $\{a_1, a_2, ..., a_n\}$:

$$\begin{aligned} c_1 &= \{a_{k_1^1}, a_{k_1^2}, ..., a_{k_1^{r_1}}\} \\ c_2 &= \{a_{k_2^1}, a_{k_2^2}, ..., a_{k_2^{r_2}}\} \\ c_i &= \{a_{k_i^1}, a_{k_i^2}, ..., a_{k_i^{r_i}}\} \end{aligned} \tag{4.5}$$

From 4.5 the size of subset c_i is $|c_i| = r_i$, the following constraints also hold:

$$n = r_1 + r_2 + ... + r_m \tag{4.6}$$

$$0 \leq r_i \leq b_i \tag{4.7}$$

$$\begin{aligned} &c_e \neq \emptyset, c_f \neq \emptyset, e < f and \\ &\forall a_{k_e^x} \in c_e, \forall a_{k_f^y} \in c_f \quad then \\ &k_e^x < k_f^y \end{aligned} \tag{4.8}$$

The constraint 4.6 means all audio hits on the left of figure 4.15 should be allocated to exact one bag on the right side. The constrain 4.8 imposes temporal order: Given two hits

$a_x, a_y \quad in \quad H$, their corresponding allocated bag in C are c_{k_x} and c_{k_y}, then the temporal order of c_{k_x} and c_{k_y} should comply with $a_x, \quad a_y$ (function t returns the time stamp of an audio hit):

$$t(a_x) < t(a_y) \quad then \quad k_x <= k_y$$

4.5.2 Mathematical Model

The complexity of the our algorithm τ to synchronization task mainly depends on the number of allocation schemes C. The aforementioned problem can be modeled as a counting of multiset problem [79]. An upper limit of τ's time complexity is $\begin{pmatrix} m+n-1 \\ n \end{pmatrix}$.

Since the number of audio hits in a tennis match is usually thousands while the plays in a match is usually hundreds. It is computationally impossible for τ to iterate all possible C then find a best match. With domain knowledge in consideration, we adopt a cascaded solution in τ to accomplish the synchronization task. With the help of domain knowledge, TennisVis allocates detected hits into tennis sets with MSCA. Within each individual tennis set, TennisVis allocates audio hits into tennis games with MSCA. And lastly, within each tennis game, TennisVis finally puts audio hits into tennis plays.

Next we introduce the domain knowledge : the rules of timing in tennis match. According to Association of Tennis Professionals (ATP) rule book [78], the following code of conducts are enforced in a tennis match:

A.1 The time between two sets shall be within 120 seconds

A.2 The time of a changeover shall be within 90 seconds

A.3 The time between two points shall be within 25 seconds

Besides the aforementioned written rules, there are some other common rules that not stipulated in the code book.

S.1 The time between two serves of same point should be within 10 seconds.

S.2 The time between two ball hits of same server should be within 2 seconds.

According the aforementioned domain rules, we are able to develop clustering methods to allocate audio hits into "audio sets", "audio games" , and "audio plays". It should be noticed that, due to the presence of match incidents and false/missing detections of hit, it is not straightforward to split audio hits into their corresponding sets/games/points. An illustrative example can be indicated by figure 4.16:

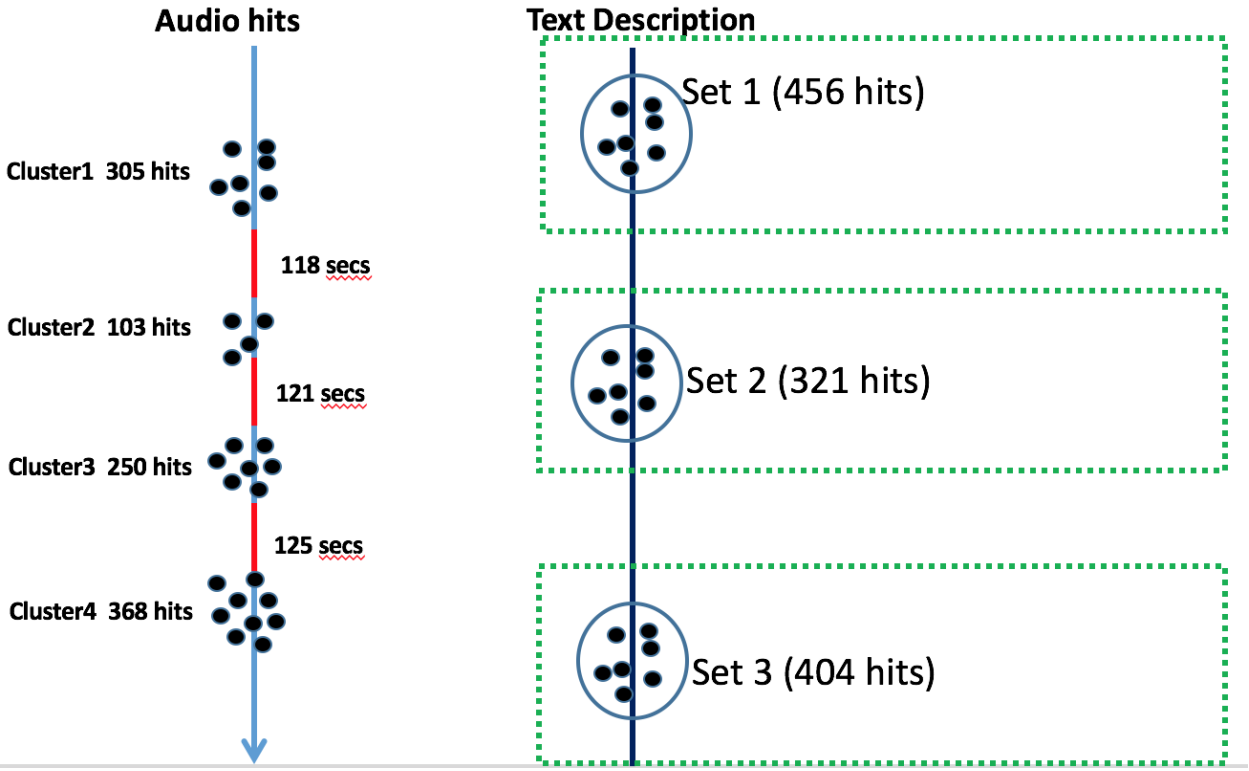


Figure (4.16) Unequal clusters due to incidents

From figure 4.16, according to the rule book, the audio hits can be split into cluster of sets. However, four clusters are computed while there are only three sets in ground truth. The error from the mis-clustering may due to inaccurate audio hit detections (missing detection due to raised noise from applause) or match incidents (medical timeouts or challenge events). In order to find an optimal allocation from $\begin{pmatrix} m+n-1 \\ n \end{pmatrix}$ possible C schedules, (In this case, m = 3, n = 4). We define a cost function φ for each allocation scheme C. Then our solution can be converted to a combinatorial optimization problem:

$$\underset{C}{\text{minimize}} \quad \varphi(C)$$

subject to (4.5), (4.6), (4.7), (4.8)

The selection of cost function φ is open, we adopt a heuristic definition in equation 4.9:

$$\varphi(C) = |b_i - \sum_{p=1}^{r_i} a_{k_i^p}| \tag{4.9}$$

Algorithm 2: MSCA

```
Function count(H, B):
    global res
    global p ← empty multiset
    global mincost
    global h = 0
    countHelper(H, B, p, res, mincost, h)
    return res;

Function countHelper(H, B, p, res, mincost, h),:
    if p is a full permutation of H then
        if cost(p) smaller than mincost then
            mincost ← τ(p)
            res ← p
        return
    for i = h, ..., length(H) do
        append h_th to i_th item to last set of p
        return if termination          // used for reduction of computation

        countHelper(H, B, p, res, mincost, i)
        remove h_th to i_th item to last set of p
```

This definition sums up all the differences (absolute value) between sum of sizes of clusters allocated to a text set and the total hits of a set. Using the case in figure 4.16 as an example, one of the allocation C^α is {{305, 103}, {250}, {368}}, so cost of this allocation is $\varphi(\alpha) = |456 - (305 + 103)| + |321 - 250| + |404 - 368| = 155$.

Next we present our multiset counting algorithm (MSCA) in Algorithm 2. The MSCA

algorithm iterates all possible combinations (n out of $m+n-1$) of subsets and returns the combination with smallest φ function.

Complexity Analysis of MSCA The MSCA algorithm iterates all $\begin{pmatrix} m+n-1 \\ n \end{pmatrix}$ combinations and always memorize the one with minimal cost (computed by $\varphi()$ function). The time complexity of MSCA is $o(\binom{m+n}{m})$ (φ function runs in constant time).

At tennis match level, m is at most 5 while n (number of "audio sets") is usually below 10. At tennis set level, m is at most 13 while n (number of "audio games") is usually under 20. With these scale of input, $o\begin{pmatrix} m+n-1 \\ n \end{pmatrix}$ is still computational affordable. Actually it took less than a second applying MSCA over our selected match on previous two levels in the case study section.

However, in tennis game level, m can be up to 30 while n can also be up to 30, in these cases, it is computationally unaffordable to apply MSCA and MSCA can not finish in reasonable time. To solve this problem, we introduce Recursion Tree Pruning Technique (RTPT) to reduce number of recursion calls of *countHelper* function in $MSCA$. Figure 4.17 illustrates an example of synchronization in tennis game level.

Figure 4.18 illustrates the recursion tree of applying MSCA on a dataset illustrated by figure 4.17 but without reduction rules. It should be noted that the notation $c1$ means "cluster 1" in figure 4.17. One of the internal nodes is represented as "[C1,]", which denotes that the current allocation puts cluster 1 in first text play. One of the leafs "[C1, C2C3]" denotes the final allocation which aligns C1 to first text play and C2,C3 to second text play. The search space in figure 4.17 can be pruned by applying the domain rule **S.2** and 4.7. By applying **S.2** rule, the internal node [c1c2,] and its subtree can be pruned. The reason is that, if c1c2 as a whole be fit into text play 1, the time elapsed between first hit of c1 and last hit of c2 is 25 seconds. However, there are only 10 hits in text play 1, combined with rule **S.2**, the time elapsed between first hit and last hit of this text play should not exceed 18 seconds. The leaf node [,c1c2c3] can also be pruned since the size of the subset of $c1c2c3$

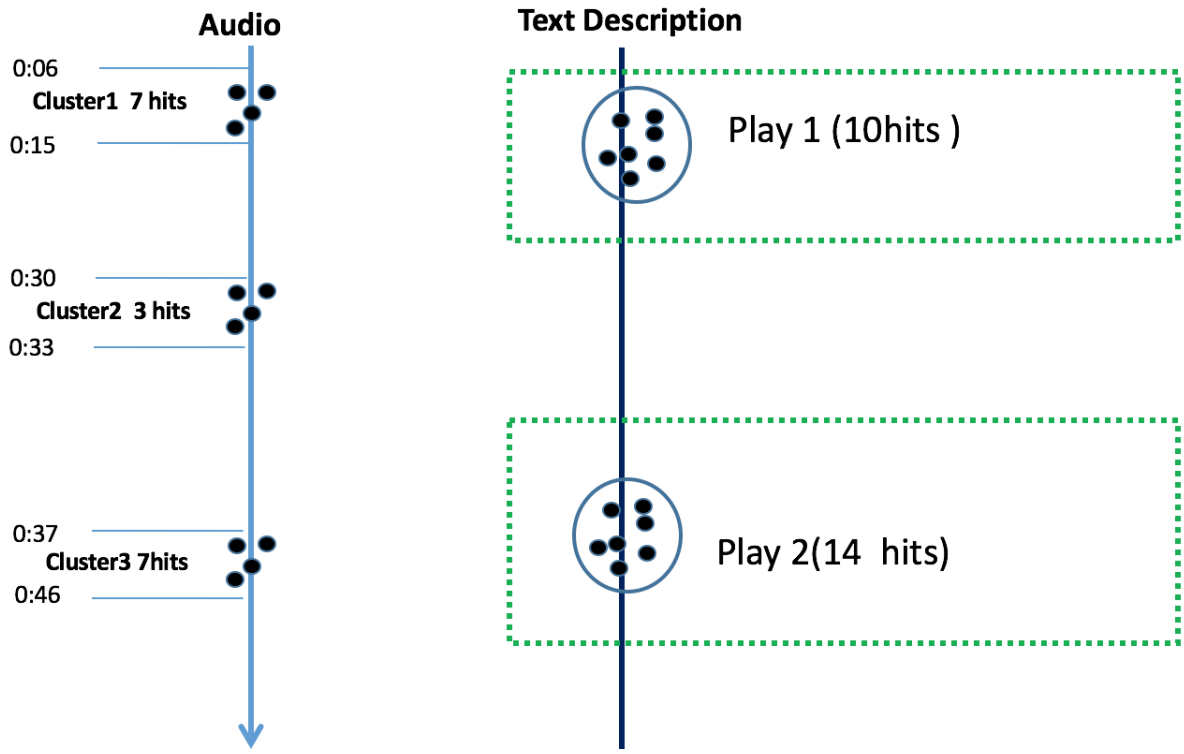


Figure (4.17) Reduction of search space

is 17 while the constraint in 4.7 imposes that this allocation can not fit text play 1(10 hits). Figure 4.19 illustrates the pruned recursion tree after applying reduction rules. The dotted branches are the pruned subtrees.

4.5.3 Rating Computation of Tennis Plays

Highlight extraction is supposed to compute rankings for different pieces in a sports match. On-site spectators are an essential part in highly competitive sports. Therefore, audio clues are important data inputs to infer the match details. For example, [80] uses audio stream to build a Hidden Markov Model to classify laughter, applause and cheer.

In TennisVis, when the synchronization task is completed, the next step is to evaluate the greatness of a tennis play. In this research, we solve this problem by computing a rating for each tennis play. The reason is that tennis play with higher ratings are more likely to be selected in match highlight. Figure 4.20 illustrates the raw audio signal of two different tennis plays and their following applauses. Play1 was first point of a tennis game and it

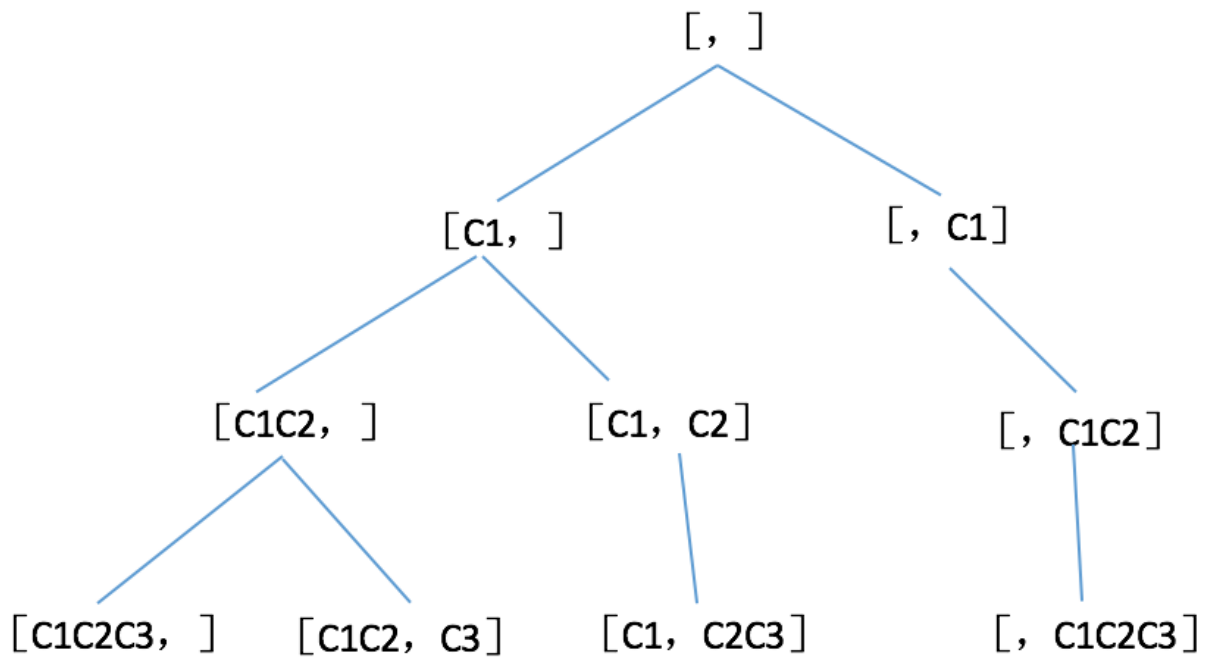


Figure (4.18) Recursion tree

ended with an unforced error. While play2 was the last point of a play with a 6-shot rally and ended with a phenomenal drop shot. The signals in the red dotted rectangles are the spectators's reactions (applauses and clapping) to these plays. It is evident that the duration of applauses can be used an indicator of the rating.

With the aforementioned idea, we developed an Energy Based Sliding Window (SBSW) technology to compute the spectators' sentimental reaction to each tennis play. We define a Short Sample Window (SSW) as a piece time series data of 0.05 second. Based on the different sample rate, the length of a window is different. For instance, the length of a SSW of 44.1k sample rate is 2205. For t_{th} SSW W^t we define its Energy of Window(EoW) as:

$$E[W^t] = \sum_{n=1}^{|W^t|} |W^t[i]| \tag{4.10}$$

We define l_t to be maximal l s.t. $\forall r \in [t, t+l], E[W^r] \geq \theta$ Then we define the rating of a tennis play P_i as follows:

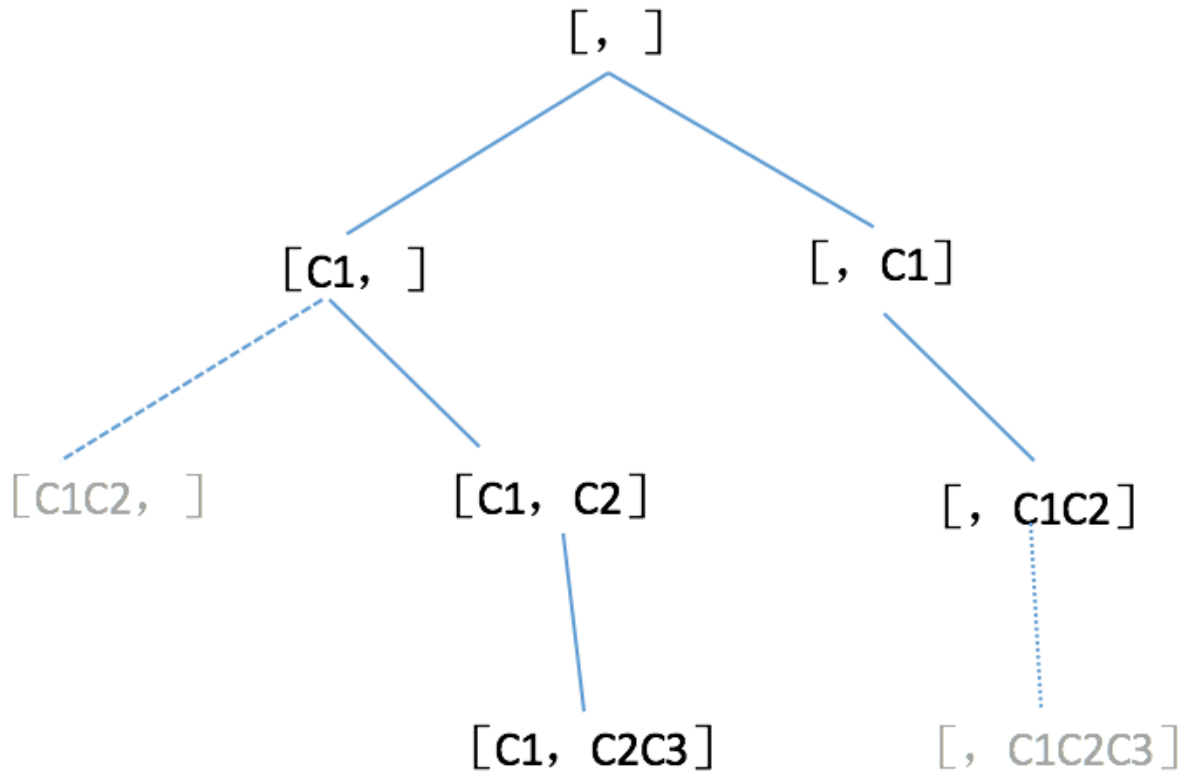


Figure (4.19) Pruned recursion tree

$$R(P_i) = \max\{ \quad l | T(P_i) \leq T(W^t) \leq T(P_{i+1})\} \tag{4.11}$$

In equation 4.11 $T()$ function returns the beginning time moment of a tennis play or a SSW. θ is a threshold picked to differentiate spectator applause and other part of the raw audio signal. Equation 4.11 defines rating of a tennis play as the length of longest consecutive SSW over a threshold θ after the time gap between that play and its next play. The design of SBSW has following advantages:

1. The time complexity of algorithm implementation is low. An upper bound of $O(n)$ is computationally affordable.

2. The adoption of SSW reduces the uncertainty from background noise.

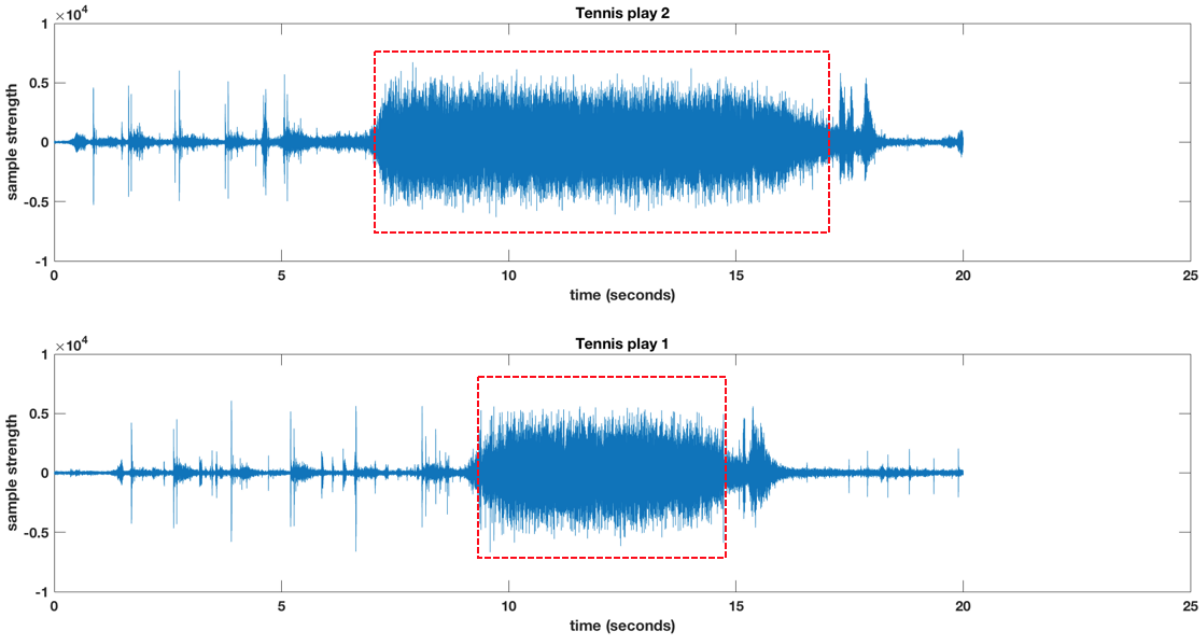


Figure (4.20) Spectators' reactions to different tennis plays

4.6 A Case Study

In this section we present experiment that verifies TennisVis algorithms and solutions. We selected men's single final of 2014 Wimbledon to demonstrate the usability of TennisVis. Our main goal here is to verify the quality of highlight extraction in TennisVis.

4.6.1 Ball Hit Detection Accuracy

As we stated before, the accuracy of ball hit detection is the foundation of highlight extraction in TennisVis. In this subsection we first evaluate the performance of CCBHD algorithm. We evaluate the performance of CCBHD in the level of tennis match, tennis set and tennis game. Matchwise, there were 1873 hits by ground truth (by mining S2STD file), CCBHD detected 1284 hits. The overall detection accuracy is 68.55%. It should be noted that there was no false positive detection in the detection result. The reason is that we took the spectator noise into consideration, all the detected hits where the spectator noise is strong will be ruled out. One side effect is that some real hits will be ruled out.

Table (4.2) Detection Accuracy in Sets

Set	**Detected Hits**	**Actual Hits**	**Accuracy**
1	310	453	68.34%
2	267	353	75.64%
3	239	343	69.68%
4	283	422	67.06%
5	185	302	61.26%
Total	1284	1873	68.55%

The worst performance of CCBHD took place in Set 5. Figure 4.21 illustrates game-by-game detection performance in Set 5. It can be seen from this figure that CCBHD performed poorly in Game 2, 5, 8. The main reasons for the poor performance are twofold:

1. More chip / slice hits took place in those games. These hits can not be easily detected since the sound strength is low.

2. In some of the plays of those sets, spectators reacted to players' wonder performances by screaming early in a many-shot rally. In these cases, missing detection rate is high.

4.6.2 Verification of MSCA algorithm

In this part we verify the performance of MSCA algorithm. We conduct experiments of MSCA in three levels: Set level, Game level, Point level.

Tennis Set Discovery with MSCA Table 4.3 illustrates set discovery result with MSCA.

It can bee seen that, MSCA failed to correctly discover the start and end time of Set 3 and 4. We checked the video and the reason of mismatch of set 3 and 4 are as follows:

1. Player one fell and got injured on court. He used multiple medical timeouts (around 120 seconds each). This conflicts with rule **A.1**, which we adopted in MSCA in match

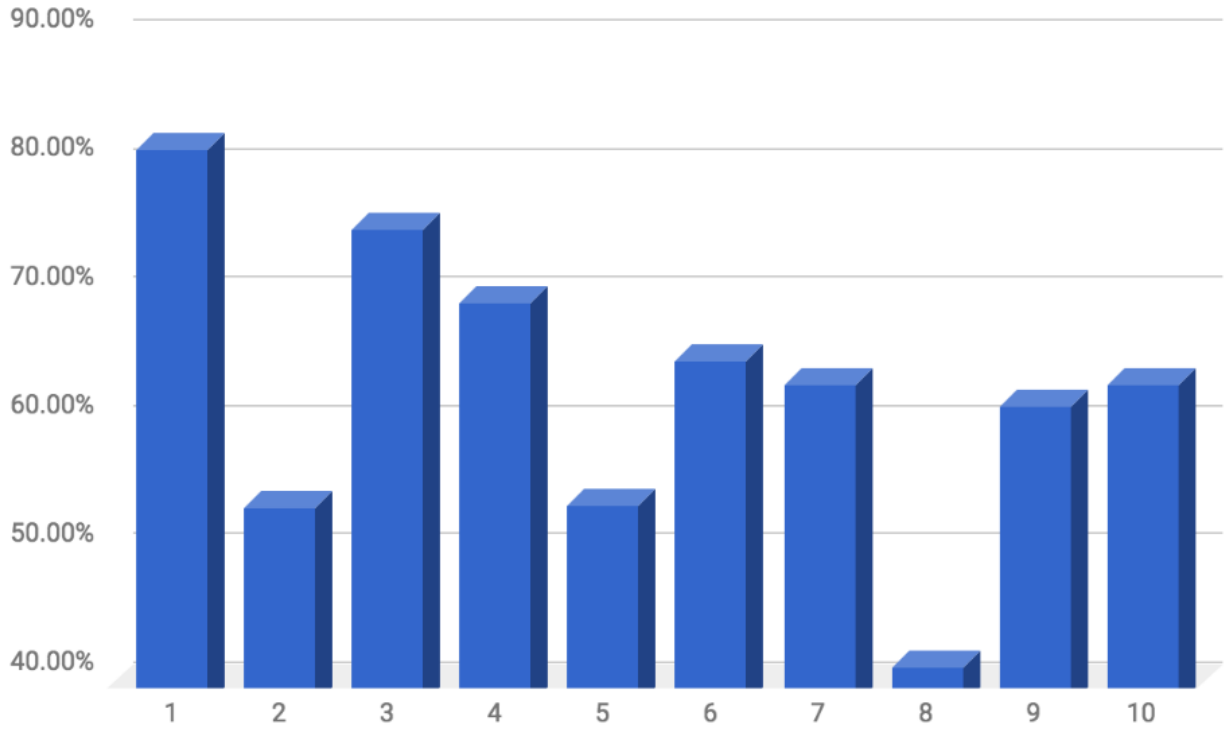


Figure (4.21) Detection performance in Set 5

level analysis. Consequently, it broke the MSCA pieces into smaller-size but more number of fractions.

2. The ball hit detection rate is not highly accurate ($\leq 95\%$).

Combined with the aforementioned reasons, MSCA failed to discover end of set 3 and start of set 4.

Tennis Game Discovery with MSCA Next we verify MSCA on tennis set level. We use Set 1 to demonstrate the performance of MSCA. The result is illustrated in table 4.4.

In general, MSCA detects all games in selected set (Set 1). MSCA failed to match game 8 and 9. Besides that, it is not "perfect match" in Game 2, 7 and 11. The deviations of end of games was due to the missing detection of ball hits. Usually spectators tend to clap or applause before last point (of a tennis game) is completely finished. Spectator noise makes

Table (4.3) Discovering sets with MSCA in a match

Set	MSCA	Ground Truth	Matched?
1	00:05 - 52:05	00:05 - 52:05	Yes
2	54:21 - 1:34:36	54:21 - 1:34:36	Yes
3	1:38:57 - 2:12:29	1:38:57 - 2:23:34	No
4	2:14:52 - 3:13:27	2:26:17 - 3:13:27	No
5	3:17:46 - 3:56:59	3:17:46 - 3:56:59	Yes

it difficult to detect end of some tennis games.

Tennis Play/Point Discovery with MSCA Next we verify the MSCA in tennis game level. Here we demonstrate the performance of MSCA's performance in two tennis games.

Table 4.6 illustrates MSCA results on Game 1 Set 1. Although MSCA identifies all plays from audio signal, the end time of some plays may be incorrect, such as play 3 (15 - 0), play 6 (30 - 15), and play 8 (40 - 15). The reason can be explained as follows: CCBHD detects 21 hits from audio signal while in ground truth there are 25 hits. 4 missing detections resulted in the inaccuracy of computation of end time of each play.

A more demonstrative case can be found in table 4.7 which presents MSCA result on Set 1 Game 9. As of MSCA input, there are 18 detected ball hits. However, there are 35 hits at ground truth side. CCBHD failed to detect ball hits after 5 shots in a 22-shot rally (Play 3), because spectators started to clapping and applause for the wonderful performance. MSCA successfully detected 4 out 8 plays in this extreme situation.

Table 4.8 illustrates the accuracy of play identification in each set. It can be seen from this table that the overall play identification accuracy is around 76.8%.

Evaluation of RTPT recursion pruning technique As we stated before, the time complexity of MSCA may be unaffordable in some extreme cases. In this part we verify the RTPT technique, which prunes the recursion tree of MSCA. We selected some games with

Table (4.4) Discovering games with MSCA in a tennis set (Set1)

Game	MSCA	Ground Truth	Matched?
1	00:05 - 2:23	0:04 - 2:23	Yes
2	3:07 - 5:42	3:07 - 6:02	Yes
3	6:33 - 8:24	6:33 - 8:24	Yes
4	9:55 - 11:45	9:55 - 11:45	Yes
5	12:19 - 13:44	12:19 - 13:44	Yes
6	15:18 - 17:00	15:18 - 17:00	Yes
7	17:31 - 19:12	17:31 - 19:14	Yes
8	20:49 - 22:33	20:49 - 23:42	No
9	22:56 - 26:34	24:20 - 27 :15	No
10	28:53 - 33:03	28:53 - 33:03	Yes
11	33:38 - 37:34	33:38 - 37:36	Yes
12	39:11 - 41:25	39:11 - 41:25	Yes
13	41:55 - 52:04	41:55 - 52:05	Yes

most m (number of tennis plays) and n (number of "audio plays") to evaluate RTPT. Table 4.9 illustrates the MSCA performance with and without RTPT. In table 4.9 , "TLE" denotes "Time Limit Exceeds". We set time limit to be 200s. It can be seen from this table that RTPT improves the running time of MSCA and it works well in the tennis match context.

Highlight Recommendation Next we present result of Highlight Recommendation of TennisVis. We use a vide highlight [81] from Youtube. It is one of most viewed highlights of the match in our case study. In ground truth side, 119 plays are offered from the highlights. Since TennisVis computes a rating for each identified tennis play, we compare the top rated 119 recommended plays to ground truth. There are round 57 plays in our recommendation are also listed by [81] , which contains 47.89% in the most viewed Youtube highlights. The overlap rate is relatively low because TennisVis failed to identify around 25% tennis points and it is subjective to select highlighted tennis points.

Table (4.5) Discovering games with MSCA in a match

Set	**# of games**	**# of matched games with MSCA**	**Accuracy**
1	13	11	84.61%
2	12	10	83.33%
3	13	9	69.23%
4	12	10	83.3%
5	10	6	60%

Table (4.6) Discovering each play(point) with MSCA in a game (Set 1 Game 1)

Play	Point	MSCA	Ground truth	Matched
1	0 - 0(Fault)	0:05 - 0:05	0:05 - 0:05	Yes
2	0 - 0	0:19 - 0:26	0:19 - 0:26	Yes
3	15 - 0	0:45 - 0:51	0:45 - 0:52	Yes
4	30 - 0 (Fault)	1:09 - 1:09	1:09 - 1:09	Yes
5	30 - 0	1:20 -1:27	1:20 - 1:27	Yes
6	30 - 15	1:50 - 1:52	1:50 - 1:54	Yes
7	40 - 15 (Fault)	2:12 - 2:12	2:12 - 2:12	Yes
8	40 - 15	2:22 - 2:23	2:22 - 2:24	Yes

Table (4.7) Discovering each play(point) with MSCA in a game (Set 1 Game 9)

Play	Point	MSCA	Ground truth	Matched
1	0 - 0	24:20 - 24:25	24:20 - 24:25	Yes
2	0 - 15(Fault)	N/A	24:45- 24:45	No
3	0 - 15	24:45 - 25:12	24:55 - 25:22	No
4	15 - 15(Fault)	25:22 - 25:22	25:51 - 25:51	No
5	15 - 15	N/A	26:03 - 26:05	No
6	15 - 30	26:35 - 26:35	26:35 - 26:35	Yes
7	30 - 30 (Fault)	26:55 - 26:55	26:55 - 26:56	Yes
8	40 - 30	27:17 - 27:17	27:17 - 27:17	Yes

Table (4.8) Identifying plays with MSCA in a match (Set-by-Set)

Set	**# of matched plays with MSCA**	**# of plays in set**	**Accuracy**
1	89	117	76.1%
2	68	92	73.9%
3	71	96	73.9%
4	84	96	87.5%
5	62	86	72.1%
Overall	374	487	76.8%

Table (4.9) RTPT recursion pruning in MSCA

Set	**Game**	m	n	with RTPT(ms)	without RTPT(ms)
1	5	5	6	0	0
1	6	8	8	0	3
1	2	10	9	0	24
1	8	10	12	3	158
3	12	12	15	1	1487
4	4	12	17	2	4258
5	7	14	16	2	13561
5	8	16	18	11	TLE
2	4	18	22	19	TLE
1	13	21	26	525	TLE

Chapter 5

CONCLUSIONS

5.1 Reading Profiling

We developed UUAT system, which collects user interaction data. We consider user click, scroll, and cursor trajectory as data source. We build a model for user reading region. By UUATs browser plugin, a users realtime reading region can be calculated. Furthermore, based on the computed reading region, more reading details can be revealed. UUAT can calculate the dwell time on each paragraph. The experiment analysis and use case analysis support our idea.

In future work, although eye-tracking has not been taken as our data collection technology, we plan to use eye-tracking to conduct verification experiments to build a more objective and accurate ground truth. Besides verification part, we plan to collect more data from each subject, such as the article reading data in a month. In personal informatics, by collecting large volume personal data, we believe there are more potential reading behavior patterns that we can study. For example, data mining and machine learning techniques can be applied so that we can explore a users reading interest once we have plentiful data, or a users knowledge acquisition can be recorded.

5.2 Data Visualization and Mining in Biological Data Exploration

CutPointVis platform enables a researcher to determine a context-dependent optimal cutpoint in a fast and con- venient way. CutPointVis provides features for a researcher to visualize Kaplan- Meier plotting and cutthrough analysis in a realtime manner. By case studies of two public datasets, CutPointVis is demonstrated to improve the research quality and productivity in survival analysis of cancer biomarker.

Although the Cox model is the pervasive optimization method in state-of- art biomarker

survival analysis, there are other methods that are also popular optimization models, such as survival. We plan to implement these optimization methods in CutPointVis to make it a more comprehensive analysis tool for biomarker analysis.

Furthermore, there are more interactions that can be conducted during an exploration process. For example, to visualize the cutpoint dichotomization qual- ity, besides the KM plot, Nelson-Aalen [82] can also be used as an reference in some situations. We plan to integrate more assistant tools to help a researcher to visualize and conclude faster and more convenient.

5.3 Tennis Visualization with On-demand Video Replay

TennisVis is visualization platform which presents match facts/statistics with brief charts. TennisVis offers query so that a use can search for tennis points according to his own preferences. Furthermore, TennisVis distinguishes itself from other similar work that, it employs an efficient Audio-based Tennis Rating Framework (ATRF), which can discover tennis play with temporal information and evaluate each tennis play with a rating. Therefore, TennisVis offers original function On-demand Video Clip Play.

For future work, we have following points to improve: 1. More domain knowledges will be introduced to improve the accuracy of MSCA. For instance, the beginning of a tennis Set has to be announced by referee. It can be used as an anchor to delimit tennis sets in a tennis match. 2. Inspired by [83], we plan to introduce sentimental analysis to help to understand how real-time match watchers evaluate the situation on courts. For instance, Twitter users tend to tweet their feelings towards a play during game/set breaks. Collective reaction of Twitter users towards specific tennis point can be summarized, such as wonderful volley from player A, great ACE from player B.

REFERENCES

[1] I. Li, J. Forlizzi, and A. Dey, "Know thyself: monitoring and reflecting on facets of one's life," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2010, pp. 4489–4492.

[2] I. Li, A. Dey, and J. Forlizzi, "A stage-based model of personal informatics systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 557–566.

[3] K. Kasemsap, "The role of data mining for business intelligence in knowledge management," *Integration of data mining in business intelligence systems*, pp. 12–33, 2015.

[4] M. Swan, "Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking," *International journal of environmental research and public health*, vol. 6, no. 2, pp. 492–525, 2009.

[5] D. Lupton, "Self-tracking modes: Reflexive self-monitoring and data practices," *Available at SSRN 2483549*, 2014.

[6] ——, "Self-tracking cultures: towards a sociology of personal informatics," in *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design*. ACM, 2014, pp. 77–86.

[7] G. Wolf, "The data-driven life," *The New York Times*, vol. 28, p. 2010, 2010.

[8] P. I. A. Godinho, B. S. Meiguins, A. S. G. Meiguins, R. M. C. do Carmo, M. de Brito Garcia, L. H. Almeida, and R. Lourenco, "Prisma-a multidimensional information visualization tool using multiple coordinated views," in *The 11th International Conference Information Visualization, 2007. IV'07.* IEEE, 2007, pp. 23–32.

[9] M. Lawrence, E.-K. Lee, D. Cook, H. Hofmann, and E. Wurtele, "explorase: Exploratory data analysis of systems biology data," in *International Conference on Coordinated and Multiple Views in Exploratory Visualization.* IEEE, 2006, pp. 14–20.

[10] M. Mramor, G. Leban, J. Demšar, and B. Zupan, "Visualization-based cancer microarray data classification analysis," *Bioinformatics*, vol. 23, no. 16, pp. 2147–2154, 2007.

[11] V. Renò, N. Mosca, M. Nitti, C. Guaragnella, T. D'Orazio, and E. Stella, "Real-time tracking of a tennis ball by combining 3d data and domain knowledge," in *Technology and Innovation in Sports, Health and Wellbeing (TISHW), International Conference on.* IEEE, 2016, pp. 1–7.

[12] H.-T. Chen, W.-J. Tsai, S.-Y. Lee, and J.-Y. Yu, "Ball tracking and 3d trajectory approximation with applications to tactics analysis from single-camera volleyball sequences," *Multimedia Tools and Applications*, vol. 60, no. 3, pp. 641–667, 2012.

[13] M. Archana and M. K. Geetha, "Object detection and tracking based on trajectory in broadcast tennis video," *Procedia Computer Science*, vol. 58, pp. 225–232, 2015.

[14] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image processing*, vol. 12, no. 7, pp. 796–807, 2003.

[15] V. Reno, N. Mosca, M. Nitti, T. D'Orazio, D. Campagnoli, A. Prati, and E. Stella, "Tennis player segmentation for semantic behavior analysis," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–8.

[16] V. Renò, N. Mosca, M. Nitti, T. DOrazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella, "A technology platform for automatic high-level tennis game analysis," *Computer Vision and Image Understanding*, 2017.

[17] X. Tong, Q. Liu, Y. Zhang, and H. Lu, "Highlight ranking for sports video browsing," in *Proceedings of the 13th annual ACM international conference on Multimedia.* ACM, 2005, pp. 519–522.

[18] H. Boukadida, S.-A. Berrani, and P. Gros, "A novel modeling for video summarization using constraint satisfaction programming." in *ISVC (2)*, 2014, pp. 208–219.

[19] Z. Zhao, S. Jiang, Q. Huang, and G. Zhu, "Highlight summarization in sports video based on replay detection," in *IEEE international conference on multimedia and expo*. IEEE, 2006, pp. 1613–1616.

[20] K. Pradeep, "Significant event detection in sports video using audio cues," *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 3, no. 1, 2013.

[21] E. K. Choe, N. B. Lee, B. Lee, W. Pratt, and J. A. Kientz, "Understanding quantified-selfers' practices in collecting and exploring personal data," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2014, pp. 1143–1152.

[22] R. Karkar, J. Fogarty, J. A. Kientz, S. A. Munson, R. Vilardaga, and J. Zia, "Opportunities and challenges for self-experimentation in self-tracking," in *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 2015, pp. 991–996.

[23] A. Rapp, F. Cena, J. Kay, B. Kummerfeld, F. Hopfgartner, T. Plumbaum, and J. E. Larsen, "New frontiers of quantified self: finding new ways for engaging users in collecting and using personal data," in *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 2015, pp. 969–972.

[24] "Patients like me," accessed: 2010-09-30.

[25] G. Buscher, R. Biedert, D. Heinesch, and A. Dengel, "Eye tracking analysis of preferred reading regions on the screen," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2010, pp. 3307–3312.

[26] R. V. Menon, V. Sigurdsson, N. M. Larsen, A. Fagerstrøm, and G. R. Foxall, "Consumer attention to price in social commerce: Eye tracking patterns in retail clothing," *Journal of Business Research*, vol. 69, no. 11, pp. 5008–5013, 2016.

[27] E. Snyder, R. A. Hurley, C. E. Tonkin, K. Cooksey, and J. C. Rice, "An eye-tracking methodology for testing consumer preference of display trays in a simulated retail environment," *Journal of Applied Packaging Research*, vol. 7, no. 1, p. 6, 2015.

[28] Q. Li, Z. J. Huang, and K. Christianson, "Visual attention toward tourism photographs with text: An eye-tracking study," *Tourism Management*, vol. 54, pp. 243–258, 2016.

[29] V. Navalpakkam and E. Churchill, "Mouse tracking: measuring and predicting users' experience of web-based content," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 2963–2972.

[30] J. Huang, R. W. White, and S. Dumais, "No clicks, no problem: using cursor movements to understand and improve search," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 1225–1234.

[31] D. Lagun and E. Agichtein, "Viewser: Enabling large-scale remote user studies of web search examination and interaction," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 365–374.

[32] A. H. Phillips, R. Yang, and S. Djamasbi, "Do ads matter? an exploration of web search behavior, visual hierarchy, and search engine results pages," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013, pp. 1563–1568.

[33] I. Li, J. Nichols, T. Lau, C. Drews, and A. Cypher, "Here's what i did: sharing and reusing web activity with actionshot," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 723–732.

[34] M. Speicher, A. Both, and M. Gaedke, "Was that webpage pleasant to use? predicting usability quantitatively from interactions," in *International Conference on Web Engineering*. Springer, 2013, pp. 335–339.

[35] R. Atterer, M. Wnuk, and A. Schmidt, "Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 203–212.

[36] R. L. Celsi and J. C. Olson, "The role of involvement in attention and comprehension processes," *Journal of consumer research*, pp. 210–224, 1988.

[37] A. Mishra and M. Verma, "Cancer biomarkers: are we ready for the prime time?" *Cancers*, vol. 2, no. 1, pp. 190–208, 2010.

[38] C. L. Sawyers, "The cancer biomarker problem," *Nature*, vol. 452, no. 7187, pp. 548–552, 2008.

[39] S. Gupta, A. Venkatesh, S. Ray, and S. Srivastava, "Challenges and prospects for biomarker research: a current perspective from the developing world," *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, vol. 1844, no. 5, pp. 899–908, 2014.

[40] R. Henson and W. Penny, "Anovas and spm," *Wellcome Department of Imaging Neuroscience, London, UK*, 2003.

[41] C. North and B. Shneiderman, "Snap-together visualization: a user interface for coordinating visualizations via relational schemata," in *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2000, pp. 128–135.

[42] R. Spence, *Information visualization*. Springer, 2001, vol. 1.

[43] M. F. de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 378–394, 2003.

[44] M. Katajamaa, J. Miettinen, and M. Orešič, "Mzmine: toolbox for processing and visualization of mass spectrometry based molecular profile data," *Bioinformatics*, vol. 22, no. 5, pp. 634–636, 2006.

[45] O. Cloarec, M. E. Dumas, J. Trygg, A. Craig, R. H. Barton, J. C. Lindon, J. K. Nicholson, and E. Holmes, "Evaluation of the orthogonal projection on latent structure model limitations caused by chemical shift variability and improved visualization of biomarker changes in 1h nmr spectroscopic metabonomic studies," *Analytical Chemistry*, vol. 77, no. 2, pp. 517–526, 2005.

[46] S. Wiklund, E. Johansson, L. Sjoestroem, E. J. Mellerowicz, U. Edlund, J. P. Shockcor, J. Gottfries, T. Moritz, and J. Trygg, "Visualization of gc/tof-ms-based metabolomics data for identification of biochemically interesting compounds using opls class models," *Analytical chemistry*, vol. 80, no. 1, pp. 115–122, 2008.

[47] A. Amaro, A. I. Esposito, A. Gallina, M. Nees, G. Angelini, A. Albini, and U. Pfeffer, "Validation of proposed prostate cancer biomarkers with gene expression data: a long road to travel," *Cancer metastasis reviews*, vol. 33, no. 2-3, p. 657, 2014.

[48] J. Budczies, F. Klauschen, B. V. Sinn, B. Győrffy, W. D. Schmitt, S. Darb-Esfahani, and C. Denkert, "Cutoff finder: a comprehensive and straightforward web application enabling rapid biomarker cutoff optimization," *PloS one*, vol. 7, no. 12, p. e51862, 2012.

[49] R. L. Camp, M. Dolled-Filhart, and D. L. Rimm, "X-tile a new bio-informatics tool for biomarker assessment and outcome-based cut-point optimization," *Clinical Cancer Research*, vol. 10, no. 21, pp. 7252–7259, 2004.

[50] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[51] W. S. Cleveland and R. McGill, "The many faces of a scatterplot," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 807–822, 1984.

[52] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems*. Springer, 1991, pp. 199–233.

[53] D. G. Altman, B. Lausen, W. Sauerbrei, and M. Schumacher, "Dangers of using optimal cutpoints in the evaluation of prognostic factors," *Journal of the National Cancer Institute*, vol. 86, no. 11, pp. 829–835, 1994.

[54] C. Contal and J. O'Quigley, "An application of changepoint methods in studying the effect of age on survival in breast cancer," *Computational statistics & data analysis*, vol. 30, no. 3, pp. 253–270, 1999.

[55] B. A. Williams *et al.*, "Finding optimal cutpoints for continuous covariates with binary and time-to-event outcomes," 2006.

[56] J. Mandrekar, S. Mandrekar, and S. Cha, "Cutpoint determination methods in survival analysis using sas®," in *Proceedings of the 28th SAS Users Group International Conference (SUGI)*, 2003, pp. 261–28.

[57] R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," *Nucleic acids research*, vol. 30, no. 1, pp. 207–210, 2002.

[58] J. Wang, S. Wen, W. F. Symmans, L. Pusztai, and K. R. Coombes, "The bimodality index: a criterion for discovering and ranking bimodal signatures from cancer gene expression profiling data," *Cancer informatics*, vol. 7, p. 199, 2009.

[59] X. He and Y. Zhu, "Tennismatchviz: A tennis match visualization system," *Electronic Imaging*, vol. 2016, no. 1, pp. 1–7, 2016.

[60] X. Zhou, L. Xie, Q. Huang, S. J. Cox, and Y. Zhang, "Tennis ball tracking using a two-layered data association approach," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 145–156, 2015.

[61] D. Demaj, "Geovisualizing spatio-temporal patterns in tennis: An alternative approach to post-match analysis," in *Proceedings of the 26th International Cartographic Conference*, 2013.

[62] H. Pileggi, C. D. Stolper, J. M. Boyle, and J. T. Stasko, "Snapshot: Visualization to propel ice hockey analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2819–2828, 2012.

[63] F. Beck, M. Burch, and D. Weiskopf, "Visual comparison of time-varying athletes performance," in *Proceedings of the 1st Workshop on Sports Data Visualization*, 2013.

[64] B. Moon and R. Brath, "Bloomberg sports visualization for pitch analysis," in *Workshop on Sports Data Visualization*, 2013.

[65] A. Rehman and T. Saba, "Features extraction for soccer video semantic analysis: current achievements and remaining issues," *Artificial Intelligence Review*, pp. 1–11, 2014.

[66] T. Saba and A. Altameem, "Analysis of vision based systems to detect real time goal events in soccer videos," *Applied Artificial Intelligence*, vol. 27, no. 7, pp. 656–667, 2013.

[67] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3584–3592.

[68] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *European conference on computer vision*. Springer, Cham, 2014, pp. 787–802.

[69] D. Tjondronegoro, Y.-P. P. Chen, and B. Pham, "Sports video summarization using highlights and play-breaks," in *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*. ACM, 2003, pp. 201–208.

[70] Z. Wang, J. Yu, Y. He, and T. Guan, "Affection arousal based highlight extraction for soccer video," *Multimedia Tools and Applications*, vol. 73, no. 1, pp. 519–546, 2014.

[71] C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Transactions on Multimedia*, vol. 10, no. 7, pp. 1342–1355, 2008.

[72] H. Sundaram and S.-F. Chang, "Video scene segmentation using video and audio features," in *Multimedia and Expo. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 1145–1148.

[73] Q. Ye, Q. Huang, W. Gao, and S. Jiang, "Exciting event detection in broadcast soccer video with mid-level description and incremental learning," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 455–458.

[74] Y.-L. Kang, J.-H. Lim, Q. Tian, and M. S. Kankanhalli, "Soccer video event detection with visual keywords," in *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, vol. 3. IEEE, 2003, pp. 1796–1800.

[75] TennisAbstract, "Tennis abstract," http://www.tennisabstract.com/charting/meta.html, 2017, [Online; accessed 09-May-2017].

[76] B. Zhang, W. Dou, and L. Chen, "Ball hit detection in table tennis games based on audio analysis," in *Pattern Recognition. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 220–223.

[77] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill New York, 1986, vol. 31999.

[78] ATP, "Continuous Play / Delay of Game," http://www.atpworldtour.com/-/media/files/rulebook/2017/2017-atp-rulebook_chapter-viii_30jan17.pdf, 2017, [Online; accessed 09-May-2017].

[79] A. Syropoulos, "Mathematics of multisets," in *Workshop on Membrane Computing*. Springer, 2000, pp. 347–358.

[80] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Multimedia and Expo. Proceedings of International Conference on*, vol. 3. IEEE, 2003, pp. III–37.

[81] ProTeox, "Match Higlights 2014 Wimbledon Men's Single Final," https://www.youtube.com/watch?v=9hdHzCVhmeg, 2014, [Online; accessed 09-May-2017].

[82] A. Winnett and P. Sasieni, "Adjusted nelson–aalen estimates with retrospective matching," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 245–256, 2002.

[83] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, 2012, pp. 189–198.